# WAR GAMING WITH A DIGITAL COMPUTER —— A DISCUSSION AND AN EXAMPLE

PAUL B. WATSON, JR.
and
THOMAS R. ABERNATHY

# WAR GAMING WITH A DIGITAL COMPUTER - A DISCUSSION AND AN EXAMPLE


by

Paul B. Watson, Jr.

Lieutenant Colonel, United States Marine Corps

and

Thomas R. Abernathy

Captain, United States Marine Corps


Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE


United States Naval Postgraduate School
Monterey, California

1 9 6 1

# WAR GAMING WITH A DIGITAL COMPUTER - A DISCUSSION AND AN EXAMPLE

by

Paul B. Watson, Jr.

and

Thomas R. Abernathy

This work is accepted as fulfilling
the thesis requirements for the degree of

MASTER OF SCIENCE

from the

United States Naval Postgraduate School

# ABSTRACT

The availability of high speed digital computers to the military and to industry has permitted the subject of War Games to be investigated in a different light in recent years. A general history, a discussion, some techniques, and an example of this new version of an old tool are here presented.

The authors wish to express their thanks to Professors Charles C. Torrance and Elmo J. Stewart for their encouragement and assistance during the writing of this paper, and to Mrs. Gwen Waddington for her speedy and excellent typing. The suggestions of Dr. Donald Guthrie of Stanford Research Institute were very helpful in the construction of the example game and in providing additional corroboration for the conclusions reached in Appendix C.

WAR GAMING WITH A DIGITAL COMPUTER - A DISCUSSION AND AN EXAMPLE

prepared by

LtCol. Paul B. Watson, Jr., USMC

and

Captain Thomas R. Abernathy, USMC

## SUMMARY

### Problem

To present in a readable and logical form the subject of War Gaming on a high speed digital type computer, for the purpose of edifying the persons concerned with the design, development or tactical employment of new weapons systems, or the training of operators for new or existing weapons systems.

### Discussion

Computerized war games are presently being employed by all of the research agencies, civilian and military, of the Department of Defense in the design and development of the future weapons of the Armed Forces. Other types of war games (maneuvers and exercises) are also employed by the operating forces and the training establishments of the Armed Forces. There exists then a need, on the part of the average officer, for a deeper understanding of the capabilities and limitations of the

war gaming techniques, particularly those of the type suitable for play on a high speed digital computer. The introduction into the higher command centers of this type of computer (NTDS, MTDS, ATDS, etc.) in the immediate future can provide these centers with a handy war gaming capability.

Conclusions

With an understanding of the Monte Carlo type of war game and the availability of such high speed digital computers, along with the trained programmers that will be assigned to operate them, commanders can effectively employ the war game - computer team to:

1) Better understand the problems of their tactical and strategic operations.

2) Develop both gross and detailed modifications to their present doctrines.

3) Aid in the training of the operating personnel, particularly their decision-making functions.

4) Gain a significant improvement in the effectiveness of the operating forces, by using the results of 1, 2, and 3 above.

TABLE OF CONTENTS

## CHAPTER I

## A HISTORY OF WAR GAMING

War games of one sort or another have been played by military and civilians alike for about as long as wars have been fought. A quotation attributed to Lao Tze, the Chinese strategist of about 300 B.C., "the commander who makes many calculations wins battles", implies that even then there existed some formalism of thought concerning new tactics, doctrine or plans for an engagement.

The ever-popular game of Chess is mentioned in the writings of Homer as being popular and old in his time. Chess probably originated in India, then migrated to Persia, Egypt, Greece, and then to Europe about 600 A.D. The simpler game of Checkers antidates Chess and was played frequently in ancient Egypt.

Little is recorded about War Games as such, prior to the 16th Century. During this time, however, many games were devised to aid the memory of the younger officers in the very formalized rules of formations for various types of troops under given conditions. In medieval and rennaissance times such instruction as was given was on a tutorial basis, with rote learning of empirical rules a mainstay of the instructional process. Attempts to make the games more realistic were frequent

at all stages of the development of the games of war.

Helwig, Master of Pages at the court of the Duke of Brunswick, in 1780 devised a major modification to the game of Chess. This game was used to train the pages for their future roles in the Army. His game board consisted of 1664 squares of different colors to depict the various types of terrain, and numbers to represent the unique features of the ground. Each side was permitted more than one player, and an umpire presided over the game. Many pieces were provided to represent fusileers, dragoons, hussars, artillery, pontoniers, pontoons and entrenchments. Minute and detailed rules were laid down for the permitted formations and movements. The board was divided into two camps by a line as the boundary between two countries. The battle action was supervised by the umpire using the rules to make the necessary decisions.

In 1798, Georg Vinturinus, a military writer and tactician at Schleswig, moved his game from a chess board of many squares to a stylized chart. This game, like most of the period, was intended for use in the military schools. Although this game was played on a stylized chart, it was similar in most respects to Helwig's; however, it had many more rules of play, and in general they were more complex. In an attempt to insert even more realism into the game, the factors

of logistics and communications were included, as was the making of decisions based upon the throw of dice. This is one of the earliest examples of the part played by chance being included in the game, recognition of the fact that all battle outcomes are not completely deterministic.

Napoleon's complete disregard for the conventional tactics of his time and his subsequent victories in Europe, led to the abandonment in many circles of the existing war games as a training device. However, Napoleon himself planned his campaigns by moving colored pins across a detailed map of the proposed battle area. This, strangely enough, is not unlike a war game.

A civilian, Von Reisswitz, the Prussian War Counselor at Breslau in 1811 made another significant change to war gaming. He transferred his game from the stylized chart to a sand table. A scale of 1:2373 was used, and the terrain features could be moulded as desired into the sand. Because in this case the movement of the troops and their equipment was not restricted to squares, a significant step forward was made in realism. The detailed rules remained.

In 1824, Von Reisswitz, Junior, then a Lieutenant of Prussian Guard Artillery, adapted his father's game to a realistic map with a scale of 1:8000. His elaborate rules were published with modifications from 1824

to 1828. This game won the approval of the Chief of the German General Staff, and was introduced into the Prussian Army as an official training aid. In this game an umpire was kept informed of the plans of both sides and he controlled the execution of them. He also acted as a source of intelligence. Each side was given a general and a special situation, and was required to issue all of the necessary and pertinent orders, in writing. The umpire displayed on the map only those elements of the opposing forces as would normally be visible to each other. Here also the chance nature of warfare was recognised by employing dice to decide the outcome of certain events. This game was more or less popular until about 1876, when Von Verdy du Vernois and others advocated the abandonment of the majority of the written rules, and the reliance upon the decisions of an experienced umpire. This umpire was to make all decisions on the results of engagements and so forth based upon his own battle experience.

This break from tradition resulted in two types of war games being played: Rigid Kriegsspiel, with the detailed rules, and Free Kriegsspiel, under the control of an experienced umpire.

These war games were introduced concurrently into the armies of Austria-Hungary, England, Italy, France, Russia, Turkey, Japan, and the United States; and this

introduction was effected simply by translation from the German publications of the existing games into the new languages.

In 1883, Major Livermore, of the U. S. Corps of Engineers, introduced his American Kriegsspiel. This was based upon the writing of Von Tschischwitz and was considered to be a great improvement. At about the same time, and independently, Lieutenant Totten of the U. S. 4th Cavalry brought forth his game, "Strategos". Both of these games were forms of the German Rigid Kriegsspiel, and they made use of detailed maps which included contour lines to indicate the terrain features. Totten's game was the more flexible of the two. Neither of these two games gained very much popularity in the Army.

In the early Twentieth Century the Free Kriegsspiel type game became popular in the U. S. Army and came to be known as a map maneuver, which was closely controlled by an experienced umpire, or director. Such games were made feasible by the new techniques of map making and reproduction. These techniques permitted maps to be produced of general areas and were not limited to historical battle sites or idealized terrain as before.

Up to the beginning of World War II, Germany made the most use of war games for training purposes. These games underwent many changes due to the changes in

weapons, doctrine, and tactics through the passing years. In 1917 the German Army High Command caused the Spring Offensive of 1918 to be played as a strategic war game. This large game showed up the low probability of success of the offensive, which, as it turned out, failed. After World War I the war game became important in the German army because of the many restrictions placed upon the Germans by the victors. War Games rated very high in their training program and embodied:

a) emphasis on decision making,

b) emphasis on order writing.

These games were played at least one day each week during the winter months, and included all echelons of command down to Regimental and detached Battalion headquarters. The Army Commanders were briefed on the situation and these briefings were passed down the line until every headquarters participating was apprised of the simulated situation. Smaller war games were used along the line to point up the situation and the decisions that had been reached by higher commands.

The initial success of the German Armies in World War II can in a large measure be attributed to their reliance upon war games in the planning stages of all major operations. It was by this means that the critical points of the plans were checked and rechecked prior to

their being carried out in the field.

In 1940 and 1941 the Japanese used war games of this type (Free Kriegsspiel) to plan their initial moves in the Pacific. Here both the Army and the Navy were involved due to the great expanses of ocean in the Pacific battle area. Even the attack upon Pearl Harbor was included in their war games. The result of all of this war gaming was the rapid and scheduled advance of their forces through Malaya, Burma, the Dutch East Indies, the Philippines, the Solomons, and the other Central Pacific islands.

War games were conducted in the United States during this period at the various service schools, but they were restricted to the teaching of tactics and not for the teaching or checking out of specific war plans. Two specific war games were introduced in the U. S. at this time by H. G. Wells and Lieutenant E. A. Raymond. Although somewhat different in their details, they were similar in that the game was played on a very large board, one about the size of an indoor drill hall. These boards had built up areas to represent the terrain features, and miniature houses, railroads, rivers, woods, etc. were scattered about in a realistic manner. Wells' game even had miniature cannon that could actually fire projectiles. These were intended to replace the umpire in his decision making function. Great

difficulties were encountered in attempting to gain the necessary degree of accuracy from these small cannon. Raymond's game substituted the throwing of specially designed dice for this function. Neither of these games gained very much popularity, due mainly to their ungainliness and their inherent expensive nature.

However, many parts of these games have been adapted recently into the more modern sand table exercises in use within the Army.

Von Neumann and Morganstern in 1944 in their book, "Theory of Games and Economic Behavior", which was based upon the earlier work by Von Neuman (1927), provided war gaming techniques with a scientific method of analysis of the decision making processes. The war game model is made to simulate the actual battle condition and then solutions or results from the model can be transferred to make predictions as to the outcome of the actual battle.

Free Kriegsspiel has advanced quite a way in modern warfare plans and training. It now takes the form of map maneuvers, live maneuvers, training exercises, logistical exercises, and communication exercises. They are extensively employed by all four of the services of the Defense Department. Each of these contribute to the training of the participants for future action and they are suitable for the testing of future

operational plans. Of course umpires and controllers are still necessary and are used. They rely heavily upon their own experience and the prepared scripts furnished them. The scripts must be prepared by highly qualified and experienced personnel.

Rigid Kriegsspiel with all of its detailed rules and complicated mathematical procedures could not be played today to any great extent without the use of the recently available high speed computers. It is being used as a research tool in the evaluation of new and proposed weapon systems, the testing of combat principles and doctrine, or for the determination of the critical values of the unknown parameters of future weapon systems.

It is this type of war game that this paper will be devoted to.

Examples of Computerized War Games

Carmonette. This war game was written by the Operations Research Office of the Johns Hopkins University for the U. S. Army. It provides for a three dimensional engagement of combined arms, that is, infantry, armor, and artillery. The battlefield, the force composition, and the tactical situation are all arbitrary. It simulates the physical characteristics of terrain, men, vehicles and weapons, as well as movement, direct and indirect fire, battlefield surveillance and communications. The data on weapon types, rates of fire and hit probabilities, rates of movement, etc. are stored in the computer in advance. A company sized game takes about ten minutes running time in the ERA 1103A (UNIVAC) computer.

The Air Battle Model. This war game was written for the U. S. Air Force by the Rand Corporation. It is divided into three parts and is probably the largest and most detailed war game in existence in the U. S. today.

Part 1 the Plan Converter. This section is capable of accepting the present war plans of the Air Force and converting them into proper form so that they can be used as inputs to the war game itself.

Part 2 the Air Battle Model. This section has seven major parts or sub-routines as follows:

a) Missile launching.

b) Bomber launching.

c) Tanker operations.

d) Bomber cell handling.

e) Attrition by defenses.

f) Target selection and reconnaissance.

g) Blast damage.

Part 3 the Output System. This section sorts, compiles, and tabulates the detailed history of all of the offensive and defensive bases, sorties launched and operations performed. All of the basic statistical computations are made automatically. The sensitivity testing of the model parameters is integral to the program. Running time is equal to real time. It employs an IBM 709 computer and 12 on-line magnetic tape units.

NEWS, The Naval Electronic Warfare Simulator. This war game is somewhat unique in that it is played on an analogue type computer. It is a special purpose game and is housed in a three story building. Command centers are provided for the various commanders, and each is capable of handling from one to four forces. Each force may be anything from one ship to an entire task force. A maximum of twenty-four forces per side are available. Standard orders are issued in the

command centers and are fed to the computers which, after the necessary calculations, display the results on the umpires board, and also the pertinent information in the command centers. When weapons are fired, this information is sent to computers which calculate the resultant damage and adjust the speed of the effected ships. The human umpires can interrupt the action at any time and interject their own desires into the problem. Unlike most digital computers, NEWS runs on real time, i.e., there is no speed-up of the action.

## CHAPTER II

## SIMULATION IN WAR GAMING

General.

Readers who are familiar with simulation and model building methods may omit the reading of this chapter without loss of continuity.

War games have been in existence for several centuries, and date back to the game of Chess (see Chapter I, History of War Games). Chess, and its simpler version Checkers, cause the two players to pit their wits against each other in a game loosely based upon the battlefield situation. By the use of a stylized set of rules, men of varying capabilities are moved across a gridded board, battles are fought, and the game results in a win, loss, or draw. Because of the simplicity of this type of game very little application can be made to modern warfare. The Prussian military advanced various games of war attempting to train tacticians during the long cold winters prior to World War II. These games were similar to modern map maneuvers and exercises and required the services of highly trained umpires, completely familiar with all of the detailed and complex rules of the game. These war games have been adapted in many ways by various armed forces of the world and are used to great

advantage in the training of future commanders. Map maneuvers and exercises are presently used as training devices at all of the military education centers in the United States.

The Model. In war gaming, a model of the real world situation to be represented by the war game must be constructed. This model must depict the true situation as accurately as possible. The model is an overall quantitative picture of the operation. As such, it will require considerable work on the part of the war gamer to understand all that happens during the real battle. Three steps can be laid down for preparation of the model:

1) Define accurately the purpose of the operations.

2) Define, in detail, the measure of effectiveness to be employed to test the model.

3) Experiment with the model to find its weak points, and rewrite it to cover these points adequately.

Purpose. Before a war game is written, considerable preliminary background work must be accomplished. The capabilities and limitations of all of the engaging forces and their equipments must be thoroughly understood. All of the scientific principles involved in

the operation of the equipment must be grasped. Some examples of this might be helpful to the reader. If visual sighting is to play a role in the game, then human search patterns, the detection patterns of the human eye, and the scope and context of the orders to the lookouts must be understood; similarly for radar, sonar, and other detection systems. The ships, aircraft, and/or vehicles and their armaments must also be thoroughly analysed. Considerable effort must be given to the understanding of the opposing forces. This is commonly called the evaluation of the threat. Incidentally, this part of the study can prove to be the most difficult, since our potential enemies are not too free with information of this sort. What is generally done, for lack of better information, is to assume that the enemy has a capability at least as good as our own and sometimes better.

This part of the problem may well involve the study of several experts for a period of up to a year or so, depending upon their previous knowledge in this field.

This basic research into the problem allows the war gamer to construct the mathematical model. This is done by first writing, in gross detail, a scenario of what is expected or is likely to happen when the forces engage. This write-up must cover all aspects of

the engagement, including the unusual. From this write-up, the programmer can start to draw the necessary flow charts, and from them write the program in machine language. The model is called a mathematical model because the written description has to be converted into the mathematical equations and expressions of movements, decisions, and rules. These have to be converted into simple arithmetical or logical operations, since these are the only operations permitted on the computer. Although they are referred to as "mechanical brains", they are really quite stupid. The high speed digital computers of today have a repertoire of about 60 instructions, yet internally they are capable of only answering a question, yes or no.

Measures of Effectiveness. The choice of the measure of effectiveness, for a computerized war game, depends upon the mission of the forces that are engaged in the game. Generally they will be simple totals or ratios of some sort. The total type has been adequately covered elsewhere in this paper (See page 44). Some other examples of ratios will be given here for the further guidance of the reader. It must be remembered that the choice of the measure of effectiveness depends entirely upon the situation of the particular game, its missions, objectives, and on the particular use to

which the game is being put; therefore generalities are difficult to make. Some examples follow:

1) the difference in blue and red force casualties, divided by the remainder of the blue force.

2) the simple ratio of the two forces' losses.

3) the simple ratio of the two forces' casualties.

4) same as 1, 2, and 3 above, except that the number is calculated for a fixed percentage of casualties, for the blue forces.

5) the actual results, divided by the theoretically possible, or desired results.

6) the ratio of the percentages of casualties on each side.

7) the ratio of the casualties to the initial number, for the opposing force.

The flow charting will start in very gross style, covering only the broad aspects of the problem. As it progresses the flow charting will become more complex and detailed, until at last every detail of the model has been converted into one big, massive flow chart. This final flow chart will be converted into machine language. The process of flow charting a complicated war game, such as a fleet air defense game, involving

60-80 Naval vessels,

160 Merchant ships,

50 Attacking bombers, and

Electronic Counter Measures,

can consume as much time as four man-years. However, this time can be considerably reduced if the flow charters are capable of writing programs using the newer symbolic programming systems or compilers, such as NELIAC or FORTRAN. These are computer operated programs that have the capability of translating abbreviated English and arithmetical symbols into machine language. Some complex games will require 20,000 to 30,000 instructions.

Experiment. When the machine language program has been completed, the game is run a few times to see that it really works as it was designed. One wants to be sure at this time that all of the scheduled events can occur, and that those events that are impossible in real life do not occur in the war game. It may occur that certain unthought of events are desirable, and these events can be incorporated at this time.

Caution. The war gamers are now prepared to run the game, and commence the analysis of the outputs. This process may involve a large number of individual plays of the game. Suppose, for example, that a

particular game has sixteen parameters of interest. It is decided to vary each of them through its possible range of values, taking only the maximum and the minimum value of each. This task would require 65,536, or $2^{16}$ runs to give a sample of only one play, for each of the possible combinations of the parameter values. Suppose further that, for statistical reasons, it has been decided that a sample of thirty (30) plays for each value of the parameters would be sufficient to give adequate reliability of the mean and variance of the output. This would require 30 X 65,536, or 1,966,080 plays of the game. Further, this game is to be played on a very high speed computer, and it takes only one minute per play. The program we have outlined would require over four years of machine time, twenty-four hours a day, to complete, and this does not allow time to make the necessary parameter value changes.

Let's look at another case. Suppose our game has, as before, 16 parameters. This time, we are fairly sure of their values. We are willing to believe that we know the values they will assume 95% of the time. If we run a war game using these values, we have computed the results of a game that will occur only $0.95^{16}$, or 43.95% of the time. If we are sure of the values of the parameters only 90% of the time, the

results of the game will occur only 18.49% of the time.

<u>Testing and adjusting</u>. Assuming that the battlefield condition has been represented by a faithful image, which we have called the mathematical model, and an acceptable measure of effectiveness has been selected by which the model can be tested, two further operations must be accomplished before the model is ready for use. The model must be tested and adjusted.

The testing of the model lies in a comparison of its predicted results with the results of actual battles of the type to be evaluated. These actual battle results can be obtained from historical studies or from experiments. The experiments can run the gamut from full scale maneuvers or fleet exercises to small unit exercises. They may even be Proving Ground type tests of a given weapon. In some cases, where the weapon system under examination in the war game is a proposal for future procurement, this phase of testing the model can only be subjected to intuitive types of tests, i.e., are the predicted results approximately of the order of magnitude that an experienced person would expect. In any event, it is necessary that the results of the model compare favorably with what is expected to occur under actual conditions. The adjusting of the model goes on concurrently with this

testing. Any deficiencies in the model are corrected as they are found.

Sensitivity Analysis. After the model has been tested and adjusted, as outlined above, it is ready for final programming into machine language. The completed war game can now be manipulated, by varying the parameters of the model, to determine the sensitivity of the output to these parameters. This type of information can show whether or not a proposed system is acceptable, or what the values of the system will have to be, in order that it be acceptable. Since, in most cases, some of these parameters will be in the form of assumptions, this type of analysis can be classed as a sensitivity analysis of the input assumptions.

The various decisions or courses of action of the decision makers in the war game can also be varied. The results of this type of analysis will aid in the determination of the optimum or best course of action, tactic or strategy, to be taken in the employment of the weapon system under investigation.

Training, Uses. A completed war game, in proper form to be played on a digital computer, can be utilized to train personnel in the decision making process. The game should either be written specially for a training mission, or an existing game can be modified, to accomplish this purpose.

The game would be written in much the same manner as an analytical type of game, except that the major decisions of the game would be left out; and some type of display of the situation leading to the necessity of the decision would have to be provided.

Each of two trainees* would be provided with a General and a Special Situation. These situations would dictate the job of each trainee in this war game; i.e., he is the commander of some particular unit engaged with the enemy in some specific manner, with an objective or mission spelled out in as much detail as he would normally be expected to have. Of course the two trainees would command opposing units in the game. It is not necessary that they be in command of the highest echelon present in the game. The part of other commanders, including the superiors of both sides, would be taken by the computer.

After receipt of the General and Special Situations, the two trainees would be given a period of time to decide upon the necessary plans and orders for their initial moves. These would be handed to the computer programmer who would translate them into machine language, and insert this information into the program for the computer. A set of standard sub-routines,

*one would have to use foreign tactics

to accomplish the usual procedures, could be provided the operator of the computer. The computer would then be started and allowed to run until the first meeting of the opposing forces occurred.

At this time, the computer would be programmed to stop and print out, in standard message format, the information about the meeting engagement that each trainee, as a commander, would normally receive. Any changes in the original orders desired by the two commanders would then be accepted by the operator, inserted into the computer, and the game restarted. This process would continue, stopping to send messages from higher and lower echelons of command to the two commanders. The two commanders would continue making decisions, in their normal role, until the battle came to an end with the annihilation of one force.

If the training mission warranted, it would be possible for the two commanders to be augmented by a standard staff. In order to complete the realism of the problem, the commanders and their staffs, if present, would have to be placed in separate rooms and would communicate with the computer operator by message.

It is possible to design special equipment that will accept information directly from the computer and display it in the form of pips on a radar scope or similar screen.

Such a device could, with proper preliminary steps, run platoon commanders through a full company-sized problem at the rate of about two per hour. Similar jobs could be done for higher echelons of command and for other services.

Dr. Donald W. Meals, Director of the Combat Operations Research Group at Fort Monroe, Virginia, in his paper, "Trends in Military Operations Research", in the March - April 1961 Journal of the Operations Research Society of America, comments favorably upon the war gaming technique. He said, "War gaming is rapidly becoming one of the most significant modern tools for the study of proposed military organizations and weapons systems". He goes on to say that war games can be used to "generate synthetic history", and users can learn much by experimenting with a properly devised game. There are, however, some limitations to present-day war games, which he hopes can be obviated in the future. These are basically: the size of most realistic computer programs for modern war games, and our inability to represent decision functions as nicely as we represent mathematical functions.

<u>Summary</u>.

War gaming with high speed digital computers is a costly endeavor, and should not be entered into lightly. However, its costs are many times less than the costs

of conducting full scale live exercises, and when properly applied, can be inexpensive in the long run. In addition, for some situations that should be investigated, the equipment to be employed does not exist. One still wishes to know approximately how it will perform, or if a given proposal will be at all useful. War gaming in this case could be thought of as insurance against the procurement of obsolete or useless weaponery.

A completed computerized war game can be used to develop tactics or doctrine for the employment of future equipment or even to train the users of the equipment before it becomes available.

# CHAPTER III

## A GENERAL DISCUSSION OF COMPUTERIZED WAR GAMES

### General.

With the advent of high speed computers and their ability to make rapidly and accurately the necessary mathematical calculations, it becomes possible to program these computers to engage in all types of battles. By utilizing the techniques of the Monte Carlo Method (see Appendix A), the laws of probability or chance can be brought logically into the game without relying upon stylized rules of decision. It is well known that the outcome of a particular battle depends upon many factors, most of which are further dependent upon chance, and are not in any way controllable by the commanders at the scene. By the use of a pseudo-random number generator (see Appendix B) the elements of chance can be brought into play, and the outcome of an engagement can be made to accurately simulate actual occurrences.

### Plan.

It is the intention of this paper first to develop a simple war game, involving both an attacker and a defender operating on a reasonable piece of terrain, and then to expand this simple war game by relaxing the assumptions and allowing more complex movement and

decision procedures. For users of the game, guide lines will be laid down for further expansion and greater realism. This game will by no means be a complete and all-inclusive game; it is intended only to point out a method and a procedure. It shculd be borne in mind that this game is being written for a particular high speed digital computer (see Appendix C). The computer herein employed is the one that happens to be locally available, the Control Data Corporation's 1604. It is hoped that sufficient generality can be included in order that the flow charts appended may be programmed for other suitable computers.

The plan is to represent a given piece of terrain as a 100 x 100 matrix of some 10,000 numbers. A semi-random series of numbers will be used to specify features of the terrain; this avoids the problem of examining in detail a map of a real piece of terrain, and the necessity of having to punch 10,000 to 20,000 IBM cards. The addresses of the elements of this matrix within the computer will correspond directly to their physical locations on the ground. The information that is stored at a given address within the computer will be the altitude of the particular grid square in question, referred to its lower left hand corner. Additional information will include indicators of the type of terrain to be found at that point, its

cover, and its mobility potential for various types of vehicles. The defensive installations will be called Anti-Aircraft guns, and the attackers will be called Helicopters. However, these names are unimportant in this application, since the user may easily change them by changing the application of the game and its decision procedures to suit his own situation. Initially the helicopters will be allowed only straight-through north-south flight paths. Modifications will permit radical departures from this path as defenders are encountered.

### Uses of War Games.

War games, including map exercises and maneuvers, have been used extensively in the past as training devices for all elements of command, and for pointing out to commanders the weaknesses of planning, logistics, tactics and decision procedures. The war game that is most extensively employed by U. S. forces at the present time is particularly inept in showing up deficiencies or strong points in weaponery. This is primarily due to the poor simulation of live firing of the weapons resulting from strict safety precautions. The decisions of the umpires will cause the battle to be won or lost according to their own past experiences and the prepared plan of the maneuver. Therefore it

becomes necessary to predict strong or weak points in the weaponery, and to have decisions made as they would be made on the actual battlefield, that is, by chance in accordance with certain probability distribution functions. These functions cannot be obtained from any known type of war game. They must be obtained from detailed study of past actual engagements where that weapon was employed against the enemy, or from the tests and evaluations of new equipment. These tests are carried out by the proving grounds and by users. For instance, it is necessary to have in any war game such information as the conditional probability that a man armed with a given type of weapon will kill a given type of target under the conditions that

1) he has seen the target,

2) he can bring his weapon to bear upon it,

3) the target is within range, and

4) the projectile or missile will function properly during its entire flight to the target.

By designing an adequate experiment or series of experiments, such conditional probabilities can be computed or estimated with fairly high degrees of accuracy. With these probabilities the computerized war game can, by the use of the Monte Carlo Method (see Appendix A), predict the outcome of such engagements under the varying conditions of the situation.

The type of war game of specific concern to this paper is of the Rigid Kriegsspiel type, (see Chapter I) with the probabalistic element of chance included along with the ever present detailed rules of decision. It must be constantly remembered that, in order to be effective as either a training aid or a research tool, the game is a simulation. The degree to which this simulation duplicates the real world situation is a measure of the effectiveness of the game.

With the advent of the internal combustion engine, armor plate, electronics, guided missiles, pre-packaged rations, satellites, etc. over the past half century, the complexity of modern warfare has increased many fold. If Helwig or Von Reisswitz were to attempt to update their games to modern times, their rule books would become volumes instead of merely large pamphlets. Particularly, their concepts of space and time would have to be changed by many factors of ten. Not only have events been speeded up but also the nature and complexity of the weapons systems employed have been fantastically complicated. This speed up and complexity are now so great that it taxes even the best strategists' ability to cope with the advancing situation. Computers are being designed to help this situation. But many more decisions have to be made, with little or no time for contemplation of the possible results.

All of this adds up to the fact that not all war situations are amenable to computer techniques for analysis. Modern computers have the capability to solve some decision problems on a rather complex scale in thousandths of a second, and it is this fact that will allow us to employ them in the analysis of some battle situations.

<u>Necessary characteristics</u>.

The battle situation to be analysed by a computer must have certain basic characteristics. First, it must be of reasonably small area or context. This is to insure that the running time of the computer, for a single play of the game is short. Since many plays of the game must be made to analyse properly the probabalistic effects of the results. Second, it must be well enough understood, for the rules of engagement of the forces to be adequately represented in logical or probabalistic form. Third, it must be reducible to the form of a mathematical model that simulates the actual conditions. The degree of which all of this can be accomplished is a function: a) of the size and type of computer to be employed (see Appendix C); b) the time and effort (money) that can be expended on the solution; c) and the simplifying assumptions that the user is willing to accept, and still believe that

the predictions of the model are approximately valid.

In order to do this, the war gamer must know certain factors concerning the battle conditions:

1) the environment of the battle,

2) the nature of the opposing forces, their equipment, and their capabilities and limitations,

3) the decision rules that will be employed, and

4) the objective or mission of the two forces.

Let us examine these factors in greater detail and see more specifically what types of information are needed.

<u>The environment of the battle</u>. One needs an accurate description of the area in which the battle will be fought. This can be as simple as a stretch of ocean or as complicated as a section of the earth's surface including the bottom of the ocean, the ocean itself with varying depths, the shore line, the beach area, the inland area and the air above all of these. If one restricts the game to only two dimensions, the environment can be easily represented by a simple grid overlaid on the area. Locations can be referred to by the nearest intersecting grid lines. If one now wishes to introduce a third dimension to the problem he must correlate this dimension to the previously

mentioned grid intersections. The grid system most commonly used is the simple square grid of known and constant size. This grid size is of primary importance since this, coupled with the memory size of the computer, will determine the area over which the battle may be allowed to run. No move of any of the combatants can be computed for any distance less than the distance between two grid lines, and all of the features of the terrain within a single grid square are lumped together as the average or the pertinent feature found within that grid square. The size of the smallest unit that can engage the enemy and still have a significant effect on the overall battle is another factor that must be considered in this choice. If the grid size selected is too large the advantages of small terrain irregularities, or clumps of bushes, etc. to individuals will be lost. However, generally speaking where large units are being considered, these factors will be unimportant in relation to the size of the battle and the area covered by it. The memory size of the computer is a controlling factor on the number of grid intersections that can be stored. Suppose it is possible to store a 100 x 100 matrix (10,000 intersections). If the units engaged are the size of an infantry division, then each grid square could very adequately represent a 100 yard square, giving an

overall battlefield of 10,000 yards on a side. This seems to be sufficient under most conditions for an engagement of infantry divisions of short duration.

Grid systems are not necessarily restricted to the simple square pattern. Any pattern that can be conveniently used is acceptable. A hexagonal pattern has been successfully used, and for some applications of this pattern the computations are simpler, since the distance between the centers of adjacent hexagons is constant no matter what direction of travel is chosen. This is not the case when square patterns are employed; distances along the diagonal are longer by a factor of the square root of two (1.414) than they are along the axis of the pattern. Unless compensated for by the program, this fact could allow a bias to enter the problem since elements would travel faster along the diagonal than they would if moves were computed only on a square to square basis.

If memory storage of locations is critical, it is possible to approximate terrain and other features by ellipses and store only the necessary constants to compute the extent of the ellipse when it is needed. In this case such a formula would take the form of $x^2 + cy^2 + dxy + ex + fy + g = 0$, where c, d, e, f, and g are the constants to be stored in order to reproduce the ellipse. To compute the ellipse, it is

necessary to make seven multiplications, and this could prove costly in running time of the computer.

If the battlefield has other features over and above these three dimensions mentioned, such as: water, air or land; or roads, woods, grass, swamp, rivers, cities, etc., then this information must also be associated with the grid intersections. These environmental conditions can be stored in the computer in several ways. One method might be to use the storage address within the computer to refer to a particular grid intersection, then at that address store all of the other required information in the form of coded digits. This method requires that the memory word be unpacked each time that a certain piece of information is needed. This unpacking process is rather lengthy in computer time. A modification of this method is to store at this address only the most used piece of information, and then in another memory address pack the remaining data required less frequently. In this case, the two addresses within the memory unit must be connected by a simple relationship, to make the correlation between them as rapid and as foolproof as possible. To decide which moves the unit may make, the computer will use this supplemental information, along with the decision rules supplied. For example, suppose a tank is moving along a road in a northerly

direction. The computer knows the present location of the tank, so it brings from its memory this supplemental information on the grid intersections into which it is possible to move. It examines each of them to determine which one contains the road, and moves the tank to that one. The computer is now prepared to continue with the game. In following this decision rule, and using the supplemental information supplied, it is assured that the tank will not perform any unreasonable or unusual tactics.

<u>The nature of the opposing forces</u>. This type of information deals specifically with the characteristics of the antagonists and their equipment. We are unrestricted here in the types of forces that will be in opposition; we may have Naval, Military or Air, or combinations of all three if we desire. Two general types of information are needed, and these can be classified as follows:

a) how many?

b) what can they do?

Inputs. Under a), how many, one needs to know the numbers of each element in the opposing forces. These numbers are generally called inputs. They are the prerogative of the war gamer, and need only reflect current and correct tactics or doctrine. These numbers

will determine the size of the game to be played and to some extent the duration of the game. Also needed in this category is the information: are replacements allowed? And if so, at what time and in what numbers will they be made?

Parameters. Under b), what can they do, one needs to know the capabilities and limitations of the participants and their equipment. These numbers are generally called parameters, since they will be dictated by the weapons system and are not under the control of the war gamer. It is these parameters that really form the heart of the war game, for it is these values that are generally to be investigated, one way or another. Since these parameters are so important, it seems worthwhile to list in outline form some examples of the various types that can be encountered.

a) Factors dependent upon the mechanics of the weapon system under investigation:

1) Hit probabilities of the guns, missiles or weapon system.

2) Kill probabilities of the war heads against all types of targets.

3) rates of fire.

4) slew rates.

5) engagement times.

6) logistics (ammunition, fuel, rations, etc.

7) communications (the sharing of information).

b) Factors dependent upon the fact that human operators will be employed:

1) the ability to see or sense the target.

2) the ability to properly identify the opponent.

3) the ability to recognise the fact that a kill has been made.

4) morale and state of training.

c) Mobility factors:

1) reaction times.

2) possible movement to increase hit or kill probabilities.

3) evasive action.

4) stealth.

5) camouflage.

6) weather factors.

Users knowledge of the parameters. Significant differences in the interpretation of the output of the war game are possible, depending upon whether or not the preceding parameters are known or unknown. If all of the pertinent parameters of the engaging weapons systems are known, then the output of the game is a measure of the overall effectiveness of the system or systems under the conditions of the game (i.e., the

assumptions). In many cases the determination of effectiveness is the purpose of the game, and as such is of major significance. War games that are carried out under these conditions represent analyses of weapons systems. From these games one can determine optimal tactics for the employment of existing systems by varying these parameters and noting the changes in the outputs.

Suppose, however, that the system to be investigated does not exist in a complete and operational form. Here the war gamer might be willing to accept certain estimates or assumptions as to the possible values of these parameters. In this case, one would attempt to find the spread over which the parameter values will range. From this information he would determine a maximum, a minimum, and possibly a most likely value for each of the unknown parameters. Now one runs a long series of plays of the game, allowing these parameters to change from their maximum values to their minimum values, through the most likely values, one at a time. By an analysis of the outputs of these many games it is possible to estimate the effectiveness of the system. This is a time-consuming process, since something on the order of thirty to fifty plays of the game must be made for each set of parameter values. A similar condition exists when one is required to pass on the

effectiveness, necessity, or requirements of a proposed system. In any case the war game can be viewed as a form of "feedback" in the sense that results produced during the playing of the game are "fed back" in order to develop the tactics, strategy, or doctrine to be employed in the system in its final form. This is accomplished by holding the parameters fixed at their most likely values and varying the decision rules associated with the game. The war game can be used to generate data so that the critical parameters can be analysed to determine just how sensitive the results are to changes in these values. From this information, direct guidance can be given to the designers of the system in the form of goals, in order to make the system an acceptable one.

Decision rules. During the course of the play of a war game, many decisions have to be made by the computer for each of the elements of the game. The elements must perform some act in response to a stimulus: they move or remain in position, they attempt to identify a target they have seen, they shoot at a target they have identified, they attempt to evade an enemy who may have seen them, and they execute a host of other decisions of this type. In order that, no matter what happens during the play of the game, each element knows what

to do at every move,the rules to accomplish each of these tasks must be included in the program of the war game. These rules fall naturally into two categories:

a) deterministic.

b) probabilistic.

Let us look at an example of a situation, and see how it is handled in a typical war game.

An Example. Suppose a tank is told to attack generally north, remaining on the road, and to attack with his 90mm gun any and all enemy vehicles that he sees during this attack. This, then, is the general decision rule to be followed by the attacking tanks of this game. Notice that it contains both deterministic and probabilistic types of decisions. The deterministic rule is, "attack generally north, remaining on the road". The solution of this type of problem was covered earlier (see page 36). Notice that there is no chance element in this decision; the tank simply moves to another grid intersection. The one that he moves to is completely determined by the terrain and the road; no matter how many times this portion of the war game is run, the tank would always move to the same grid intersection. The probabalistic rule is,"attack with his 90mm gun any and all enemy vehicles that he sees during this attack". In this case the computer, knowing the three-dimensional locations of the tank, all of the

enemy vehicles, and all of the intervening terrain, can compute whether or not it is possible for the tank to see any of the vehicles. The degree of ground cover at the locations of the vehicles could also be included in this calculation, but this just makes the example more complicated. Now we all know that, even though it is possible for one person to see another, the sighting is not always made the first time that it is possible. It may occur a short time later, or it may never occur. Whether or not a sighting actually does take place is a function of: where the observer is looking; how good his eyes or other sighting aids are; his ability to distinguish the target from its background; and other similar factors. By means of controlled experiments in the field, using actual equipment, the probability that a sighting will occur can be computed. This probability may be a constant or a function of range, weather, training or other factors. It is this information that must be written into the program so that the computer can solve this problem. How does the computer determine that a sighting has or has not occurred? The method employed by the computer depends upon the form of the probability distribution. Take the case where the probability is a constant for all ranges and other factors. By means of the pseudo-random number generator (see Appendix B)

that is included in the program, a random number is generated. The resulting number is considered as a fraction between 0 and 1. This fraction is then compared to the constant probability; if the random number is the larger, no sighting occurs on this try; if smaller, a sighting does occur. In the case where the sighting probability is a function of range, for example, this function must be included in the program of the war game. It can be stored as a mathematical formula, or as a table of values of the probability versus range. The range between the two elements in question is known, so the computer either calculates the probability from the formula or searches the table for the probability at that range. Again, a random number is generated and compared to the computed probability as before.

The Objective or Mission of the Forces. The objectives, or missions, of the opposing forces are the reasons that the two forces are in the engagement. They can be stated in broad general terms in the model. This information will be used by the programmer in drawing up the necessary flow charts of the game prior to its being converted into machine language. This tells him the purposes and functions of the forces and their intentions. It is from this that he will obtain the

information that he needs to decide what types of data will be most used during the running of the game, and therefore just how these data should be stored in the computer's memory. Also, the outputs of the game will be dictated by these missions.

The outputs of the game usually are numbers representing the results of all of the various engagements that have taken place during that game. These can be briefly generalized as casualties. They will be totals of such items as: ships sunk, planes shot down, tanks or vehicles out of action or destroyed, guns destroyed, personnel killed or wounded, and the like. These numbers are intimately related to the missions of the engaging forces and the measure of effectiveness selected for the problem. In certain logistical war games, the outputs might be: tons of supplies delivered, tons of supplies undelivered, tons of supplies destroyed, or vehicle loads delivered or lost.

Measure of Effectiveness. This is a number by which the overall results of the game will be judged; it therefore requires a considerable amount of thought on the part of the war gamer in its selection. The purpose of the weapon system under study will, to a large extent, determine the measure of effectiveness selected. If the purpose of the war game is to compare two com-

peting systems, then special care must be made to insure that no bias is allowed to distort the measure of effectiveness in favor of one of the two systems. It must represent the factors of interest in both cases and not preclude the selection of one of the systems. On the other hand, if only one system is under consideration, then the measure of effectiveness could be one of a number of exchange ratios, or even more simply, the losses of both sides. Exchange ratios are the ratios of enemy to friendly losses, for the various types of equipment or personnel that are engaged. Ratios of this sort are quite frequently used; however, they can be not only misleading but even dangerous in some cases. For example, suppose the mission of force A is to prevent the penetration of elements of force B. Ratios or exchange rates are meaningless in this case, since force A might kill all of the elements of force B, except one; and that one is a penetrator. Here force A loses since it failed in its mission. In this case, one would desire a simple total of penetrators to measure the effectiveness of the system. Another good measure might be the ratio of successful attempts to failures on the part of force B. War games, played by industry, frequently use ratios involving dollars, since they play up the cost of systems in their comparisons. These are to be looked

at suspiciously, since costs are frequently difficult to estimate, and sometimes major costs are left out (such as training or maintenance). The totals of losses of critical items of supply or extremely expensive items of equipment can be effective measures in some cases. In short, in order to devise a proper measure of effectiveness, the war gamer must be thoroughly cognizant of both the objectives of the forces involved and the purpose or function of the individual equipments. In any case, this selection should not be hastily decided upon, but should result from considerable study of the problem.

Output Analysis. For a given play of the game, there will result a series of numbers that represent the output or results of that play. It must be constantly borne in mind that these results, by themselves, really mean very little. They are simply the results of that single operation, and are completely dependent upon the particular random numbers that were generated and used on that play of the game. They are no more significant to the overall problem at hand than is the result of a single engagement of two aircraft to the results of the entire war. These numbers represent simply what can happen in a single engagement. What is really important and desired to be known is,

what is the most likely outcome of the engagement, or better, what is the probable distribution of outcomes of the engagement. This information can be approximated by repeating the process a large number of times, and from the resultant data compute the average or expected value of the results. Another item that can be computed, and is of great importance, is the variance or the degree of variation of the results. The question then arises as to how many times must the game be played, in order to gain significant results. The answer to this question is one of the biggest ponderables in the field of statistics, and much beyond the scope of this paper. The reader is referred to the many texts on the subjects of Statistics, Analysis of Variance, Sequential Analysis and Statistical Decision Theory. It can, however, be pointed out what some of the critical variables are, and what are their interdependencies. The computer running time for a single play of the war game is one factor that plays a large part in this problem. On high-speed ($\frac{1}{4}$ micro-second per arithmetic operation) computers, simple programs may run on the order of one to five minutes per play. On the other hand, more complicated war games may take 15 or more hours per play. The cost of obtaining the data is a serious governing factor; for example, computer rentals can exceed eight dollars per minute.

The mean time to failure of the particular machine that is selected is another factor to be considered. If the running time of the game, or of the desired series of games, exceeds this value by a factor of more than about four or five, then the probability that correct answers will be obtained drops rapidly. This is due to the fact that computer failure grows more likely as running time increases.

In order to have results that will be significant, the assumptions of the model and the desired or possible accuracy of the results are other factors that must be considered in attempting to determine the number of plays of the game. It is possible, in the early stages of the analysis of a war game, to make a series of plays of the game in which the results are analysed in terms of the size of the sample (number of plays) taken. For instance, one might take a sample of ten plays and analyse the outputs for their mean and variance. Then, holding all of the inputs and parameters constant, take another sample of ten plays and analyse the new total of twenty plays. This process is repeated until the results of the analysis show that an acceptable minimum variation has been obtained. The result of this process is a determination of the sample size. This process has the advantage

that the plays of the game, to this point, are not wasted; and they can be included in the final analysis. Here both the factors of cost and desired accuracy can be adequately considered. (For some methods of analysis that are used, see Appendix B on the analysis conducted on various pseudo-random number generators).

The method of presentation of the results is all-important, since the average user of the results will rely heavily upon this presentation. This of course depends upon the type of the outputs, totals or ratios, and their use. Some methods of presentation of the results are given below, primarily as a guide for the reader.

Certainly, if a series of runs of the war game was the criterion used to select the number of runs per sample used in the final analysis, then a graph or curve should be presented to show the central tendency of the output data as a function of the sample size. This type of display is rapidly grasped by the user and, surely, is a better method than simply a long series of numbers. An example of this type of presentation is shown in Fig. 1 on page 50.

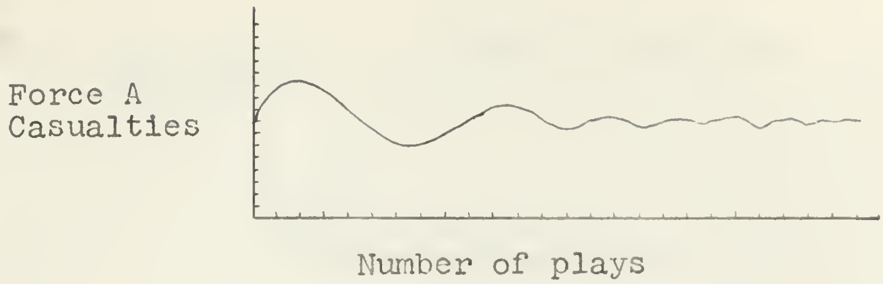



Fig. 1

Bar graphs, or curves, of the totals of the critical elements for the games run, showing their means and comparing the results for the two opposing forces, are very effective. An example of this type of presentation is shown in Fig. 2 below.

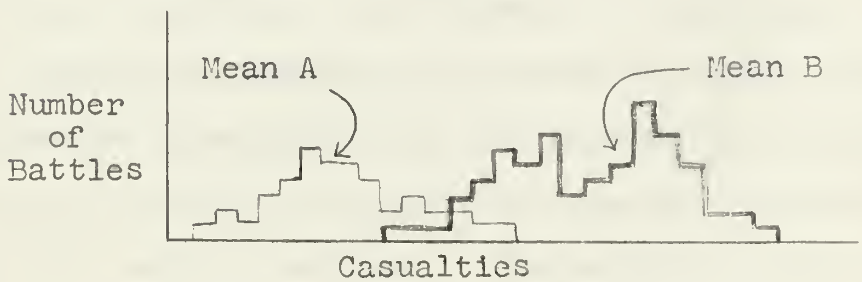



Fig. 2

Where the measure of effectiveness selected for the war game is one of the many exchange ratios, a scatter diagram is considered to be effective. An example of this type of presentation is shown in Fig. 3 on page 51.

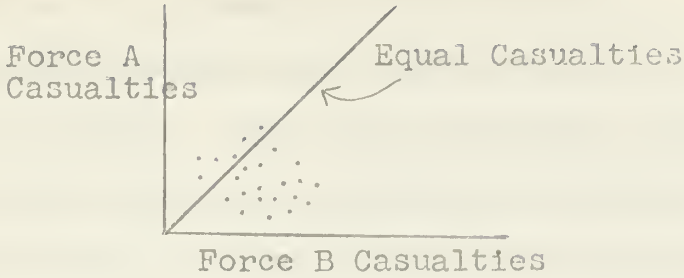



Fig. 3

The field of statistics provides many estimation procedures that may be applied to the results of war games. As a consequence of these procedures, many statements can be made as to the validity of the results. These statements generally are of the type, "the stated mean or variance is within certain bounds, with a certain 'confidence coefficient'". This means that the author has applied the appropriate statistical procedure, consistent with the assumptions, and finds that he can give the reader an interval estimate of the true mean or variance. The meaning of this interval is as follows: if this estimation procedure is repeated over and over again, a certain percentage (confidence coefficient) of the intervals will cover the true mean or variance. The confidence coefficient is usually chosen to be a fairly large percentage, say 0.90, 0.95, or 0.99. Naturally it is desirable to have the length of the confidence interval as short as possible. One way to achieve this is to increase the sample size.

It must be remembered that nearly all statistical procedures are based upon the fact that the sample drawn is random. Also many procedures assume that the sample was drawn from a normally distributed population. There do exist, however, some procedures that do not have as their basis this assumption. These tests are called non-parametric procedures. In these cases, the lengths of the confidence intervals that are obtained are, in general, larger than those for which the normality assumption is made. Therefore, if the population is approximately Gaussian, it would be desirable to use the Gaussian procedures. If, of course, the distribution deviates greatly from the normal distribution (i.e., skewed, bimodal, etc.) the probability statements obtained from a Gaussian procedure are meaningless.

The assumption of normality of the data is one that is made in most cases. It is made so often that in many cases the authors neglect to point out this fact. The reader is cautioned to look into the assumptions that are both implied and expressed, in an attempt to convince himself that they are reasonable in the light of his knowledge and experience.

Programming Techniques. There seem to be two basic methods of programming war games:

1) Time increment.
2) Event store.

Each of these methods has its own unique advantages and disadvantages; and probably because of this, only the simplest types of war games today are being programmed by purely one method or the other. The more usual procedure is to combine the two methods in a single program, in an attempt to make the most of the advantages of the two methods.

Time Increment . In this method the entire war game is divided into a series of time increments or steps. At the expiration of each time step, new positions are calculated for each of the participants on the field of battle. Using these new positions, computations are made to determine if any significant events can occur during that time step. If any can occur, then further calculations are made, for each of them, to determine what does occur. After these calculations are completed, the entire system and results of the game are updated to account for any losses or replacements. Upon completion of all of these arithmetic operations, the game is prepared to procede to the next time increment. This process is repeated until the stated battle time runs out or the casualties for one side equal the starting numbers. Rather obviously, the time increment selected is very important in this method. In deciding how long this time can be, careful consideration must be made of the

following factors:

a) The maximum rate of advance possible for each of the participants.

b) The total number of participants.

c) The number of critical events that can occur in the one time step.

If the participants have large rates of advance, as with aircraft, the time step must be made small in order that they will not move too far between checks upon their progress. If the increment of time is too large, it will be possible for more than one critical event to occur during this time interval. If one or more critical events do occur during one step, then they will have to be considered as being completely independent, whereas in fact they may very well be dependent events. For instance, in one time step an enemy unit may be sighted by several friendly units. The program would compute in each sighting what the occurrence would be. This could result in multiple kills on the same target. In the actual case, however, the sightings might have occurred serially during this time. In this case it would be possible for the first sighter to have killed the target and the remainder would not have had the opportunity to have sighted him at all.

If large numbers of participants are located within

a rather small battle area, the interactions between them grow rapidly. A consequence of this situation is to have more than one friendly unit firing upon the same enemy unit. In this manner a single unit could be killed by several units. This fact by itself is not enough to invalidate the war game,since events such as this can and do occur in real engagements. It is not always possible to determine the fact that a given hulk on the field of battle has previously been killed by our own forces, and does not constitute a threat to our side.

If the time increment selected is short, this means that it will take a long time (many steps) to run through a real time situation. Here again the cost of computer time rears its ugly head.

Event Store. In this method, the program examines each participant in turn and computes for him the next time that a critical event can occur. In other words, he is advanced along his normal path until it is possible for the event to occur. The time of occurrence of this event is computed and stored. When the entire list of participants has been exhausted, this list is examined to see which event can occur at the earliest time. The necessary calculations are then made to see just what did occur and the proper tabulations are made for the record. The event of next

earliest time is then examined in the same manner. If early events preclude the possibility of the later events, then the previously prepared list of event times must be changed accordingly. The main disadvantage of this method is that, in many cases a large amount of time is wasted in making calculations that later prove to have been useless. This method can make a significant pay-off in shortened running time for each play of the game, provided that the number of participants is not too large. A big disadvantage is the fact that programming for this method is not only more difficult than for the time increment method, but it is very time consuming. This time is expensive because it requires the services of highly trained programmers. This time can run to values of four man-years for extensive programs.

Line of sight calculations. Calculations of line of sight must be made in almost every war game that involves more than two dimensions. In the example chosen for this paper, this calculation is strictly for the human eye type of sightings. However, other associated calculations are not restricted to eye sighting alone. Calculations must be made to determine if there exist intervening obstacles to deter the effects of radar or sonar beams, or the sensors of other types of detection instruments.

There exist several methods of computing line of sight. Two possible methods will be discussed as guide lines for the reader.

Vertical angle method. In this method, the three dimensional coordinates of the two positions in question are used to compute the two angles that determine the spacial location of the line of sight. We will call one the horizontal angle and the other the vertical angle. The horizontal angle is then used to compute the first grid square that the line of sight passes over in the direction of the target. From the three dimensional coordinates of this grid square, and the coordinates of the point of origin, a vertical angle can be computed. If this vertical angle is smaller than the one to the target from the observer, then it is possible for the observer to see at least this far. This process must be repeated for each grid square over which the line of sight passes. Each newly computed vertical angle must be smaller than the original vertical angle for a sighting to take place.

Grid Line method. In this method, as before, the coordinates of the two positions are used to compute the necessary vertical and horizontal angles. From the size of the horizontal angle, the quadrant that contains the line of sight can be determined. For purposes of illustration, assume that it falls within

the second quadrant. The program moves one grid unit East and forms a right triangle with this line as its base. Using the horizontal angle, the altitude of this right triangle is computed. Using this distance, the altitude above the datum plane of the intersection of the line of sight with this North - South grid line is computed. Using this distance again, and the vertical angle, the altitude of the line of sight above the datum plane at this intersection is computed. These two altitudes are compared. If the altitude of the line of sight is the greater, then a sighting is possible. This process is repeated until the Easting coordinate of the target is reached. Next, the program moves one grid unit South from the observer, and the entire process is repeated as before. At any time an altitude is found to be in excess of the altitude of the line of sight, no sighting is possible.

General. It is possible to save running time by pre-calculations, if frequent line of sight calculations are to be made during the war game and the game is to be run many hundreds of times. In these cases, the possibility of sighting from every grid intersection to every other grid intersection is calculated and the results stored for later reference. The storage of this data can very easily exceed the capacity of the computer's memory. However, this data

can be stored in peripheral equipment (magnetic tape units, etc.) and referred to when needed.

## CHAPTER IV

## AN EXAMPLE OF A COMPUTERIZED WAR GAME

<u>The Battle Model</u>.

This particular battle model represents an engagement, in three dimensions, between a stationary defender and an airborne moving penetrator. In this model the defender and the penetrator are called anti-aircraft weapons and helicopters, respectively. These names are fairly arbitrary, and have been selected for illustrative purposes only. This model can be adapted to other types of engagements by changing:

1) the titles of the participants,
2) the assumptions,
3) the decision rules,
4) the inputs, and
5) the parameters.

<u>The Battle</u>. In this battle, several anti-aircraft weapons are placed in positions on a simulated piece of terrain. The terrain is rolling, but otherwise void of natural or man-made features. Helicopters are launched according to a probability distribution function, they overfly the terrain serially, and they are constrained to a straight North - South flight path. After each move, calculations are made to determine which

helicopters are within range of each of the anti-aircraft weapons. Using these results, further calculations are made to determine if the line of sight between gun and helicopter is free of obstacles; that is, the gun can see the helicopter. As each helicopter comes within view of the various anti-aircraft weapons, a chain of events is triggered.

First, in accordance with a given probability function, a determination is made to see if the gunner has seen the helicopter. For simplicity, it has been assumed that if he does not see the helicopter on this initial opportunity, he will not see it on this move.

Second, given that the gunner has seen the helicopter, he will attempt to bring fire to bear on it. This also is accomplished in accordance with a given probability function. Again, if he does not bring his weapon to fire at this time, he will not be afforded another opportunity until the next move of the game.

Third, having seen the helicopter and fired upon it, a probabilistic determination is made to ascertain if a hit was made.

Fourth, given all of the above, another probabilistic calculation is made to confirm the kill or survival of the helicopter.

Records are maintained of the following information:

a) the number of hits on each helicopter.

b) the number of times that each anti-aircraft weapon fires.

c) the number of helicopters that are killed.

The Terrain.

In this simplified war game, the terrain over which the battle is fought is rather unique. Since the area is to be covered by a square grid system, one hundred units on a side, there result ten thousand individual grid squares. In order to obtain the individual altitudes of each of these squares from an actual piece of the earth's surface, a considerable amount of detailed work would be required. This work, even if accomplished, would not contribute significantly to the problem at hand,which is to examine the game model. Therefore, a section of land is manufactured by the use of the "Terrain Generator". Valid results of this war game are obtainable, of course, only if "actual" terrain is used.

The "Terrain Generator" is a sub-routine of the main program. This sub-routine, by the use of randomly generated numbers and the introduction of a suitable bias, can manufacture the necessary 10,000 altitudes in short order.

Weapon Emplacement.

For the purposes of the problem faced by this paper, it is felt that randomly placed anti-aircraft weapons will suffice. Again, this is a dodge to avoid labor that does not bear on the problem at hand. This placement is accomplished by the "Place Defense" sub-routine. This sub-routine, using random numbers, scatters the anti-aircraft weapons across the generated terrain, without regard to tactics or doctrine. This dodge, of necessity, will invalidate the game results for practical application. However, if the weapons are placed in accordance with doctrine on realistic terrain, the results will be valid. This can be programmed but has not been included in the basic game. Such programming requires extensive use of decision functions derived from tactical doctrine.

Operation.

To play the game, the following inputs and parameters must be decided upon and inserted into the appropriate places in the machine language program.

Inputs.

Grid size.

Number of helicopters.

Launch interval of the helicopters.

Speed of the helicopters.

Flight altitude of the helicopters.

Track to be followed by the helicopters.

Number of anti-aircraft guns.

Number of plays desired.

New Terrain each play?

New Locations for anti-aircraft guns each play?

Parameters.

Probability function for launching a helicopter, $p_1$.

Probability function for sighting, $p_s$.

Probability function for shooting, $p_f$.

Probability function for hitting, $p_h$.

Probability function for killing, $p_k$.

Range of the anti-aircraft guns, $r_1$.

The program is then stored in the computer's memory, and the computer is started. Upon completion of the desired series of plays, the results are made available and the computer stops.

Functioning of the Game.

After insertion of the necessary inputs and parameters, the terrain is generated, and the anti-aircraft weapons are emplaced. At each move, a helicopter is launched with some probability function, $p_1$. If launched, it enters the approach lane and commences flight over the terrain. At randomly distributed intervals, successive helicopters are launched over

the terrain. The helicopters advance one grid square at a time. If the grid dimension selected is 100 meters, and the helicopter speed selected is 70 miles per hour, a time increment for the game of about 3 seconds results. Adjustments of distance can be made, and the interval selected is for convenience.

At the end of each move of the helicopters, calculations are made to determine which of them are within range of each of the anti-aircraft weapons. This information is tabulated in the "Range" table, which stores for each helicopter the weapons that can fire on it.

Using the "Range" table as a basis, calculations of line of sight are made to determine which of the guns that have helicopters within range can actually see them. The given probability function for sighting, $p_s$, is used to make the determination that a sighting can or cannot occur. This information is tabulated in the "See" table, which stores for each helicopter the weapons that now have seen it.

Using the "See" table as a basis, firing calculations are made to determine which of the guns fire at the helicopters they have in sight. The probability function for shooting, $p_f$, is used for this calculation.

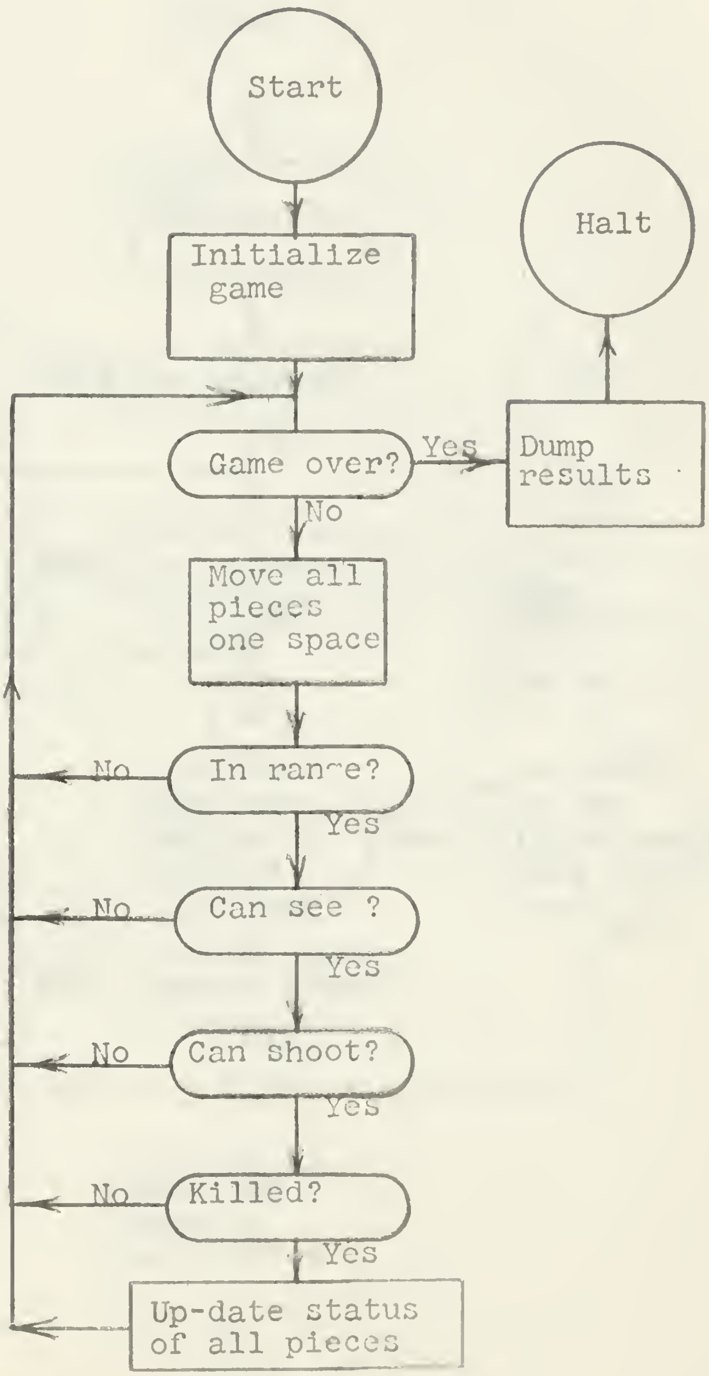
Similarly, the calculations for hitting and killing the helicopters are made, using the given probability functions, $p_h$ and $p_k$.

After the necessary deletions are made from the list of active helicopters, the program moves all of the remaining helicopters one more square forward, and the entire process is repeated.

The game ends when either all of the helicopters are killed or when the last one that has been launched passes out of the battle area.

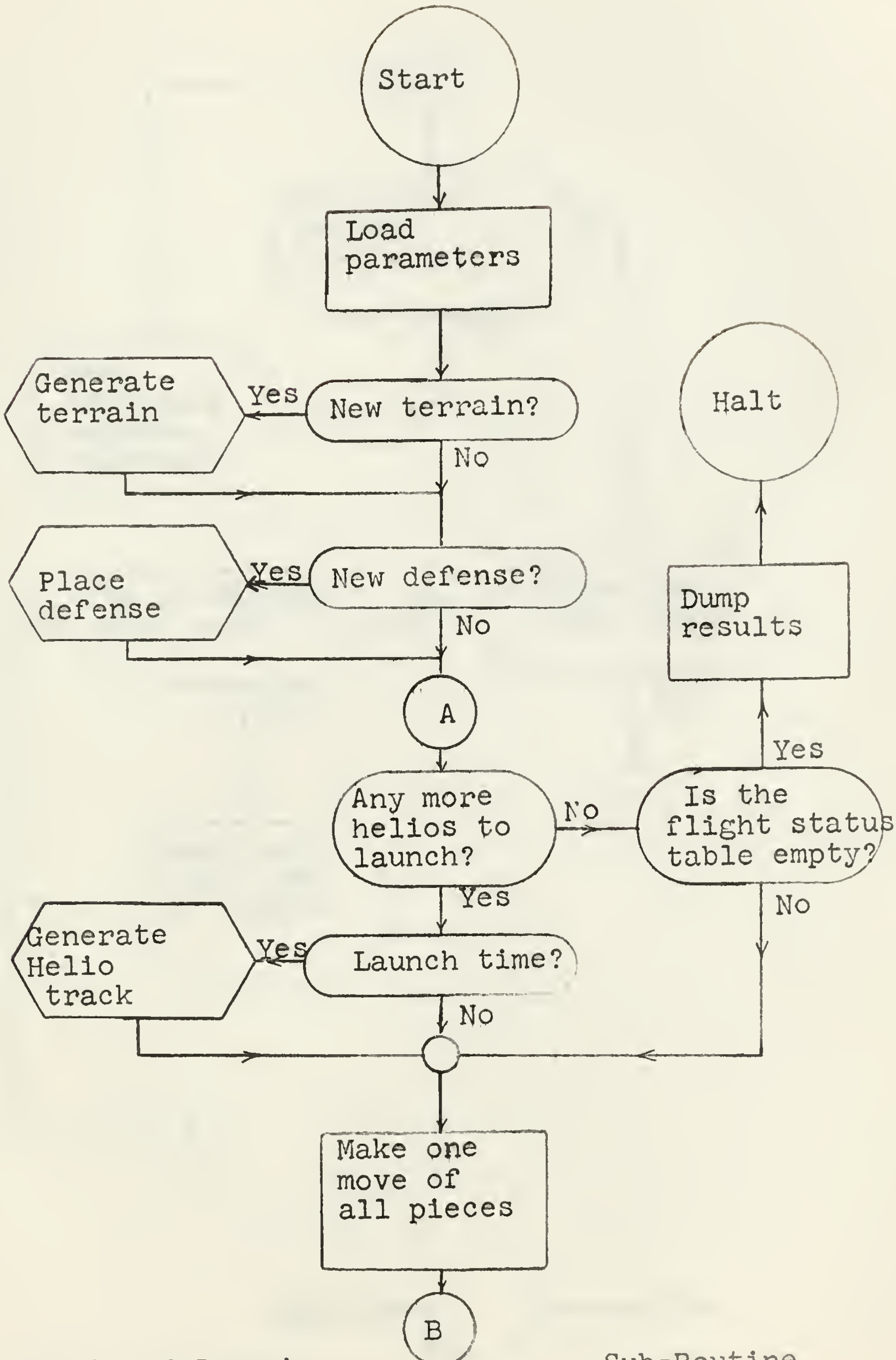
Flow Charts.

To enable the reader to utilize more properly this example game, flow charts are included in this section. Although they are not in sufficient detail to allow machine language coding directly, they can provide the trained programmer with a good starting place. The machine language coding of this example game will be available at the U.S. Naval Postgraduate School, Monterey, California. This coding has been done for the Control Data Corporation 1604 Computer. Standard symbolism has been used in the drawings of the flow charts.



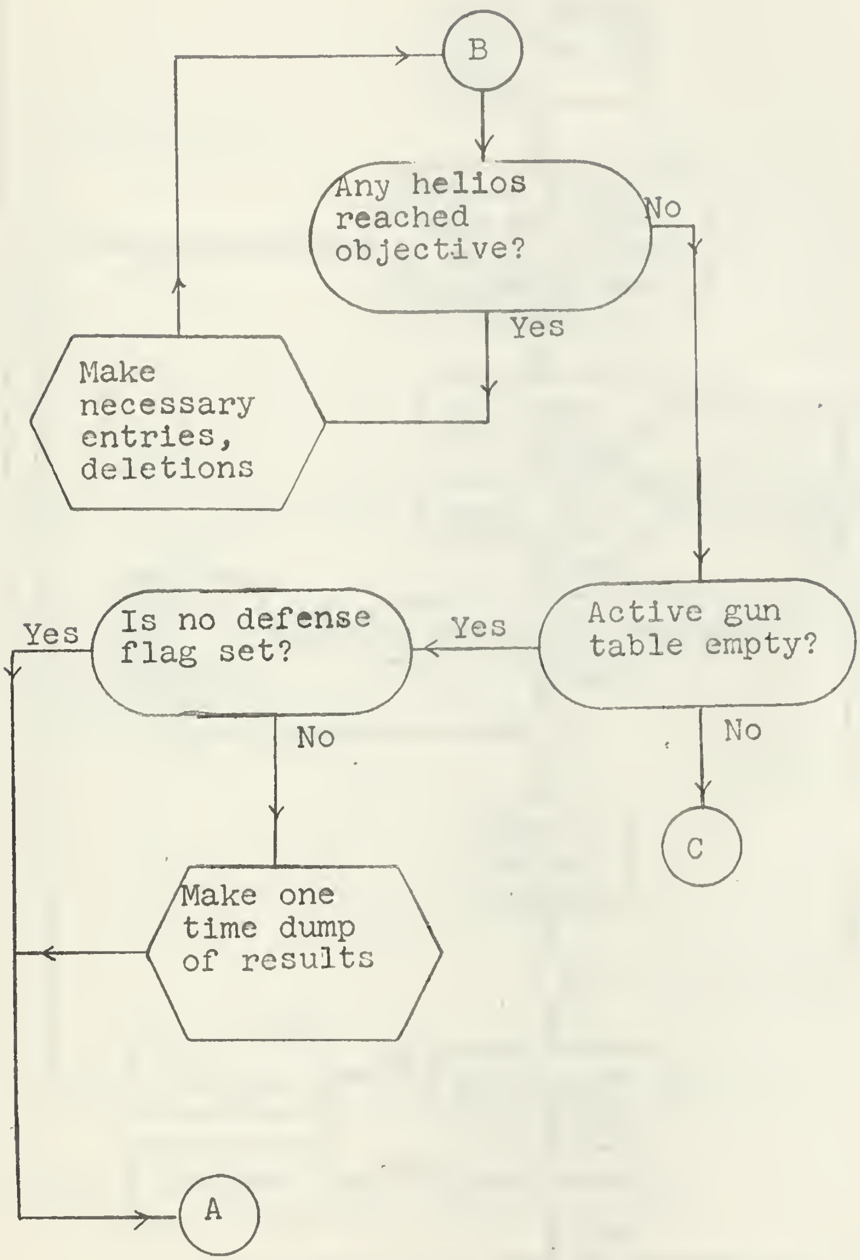

OVERALL PLAN OF THE GAME

Fig. 4

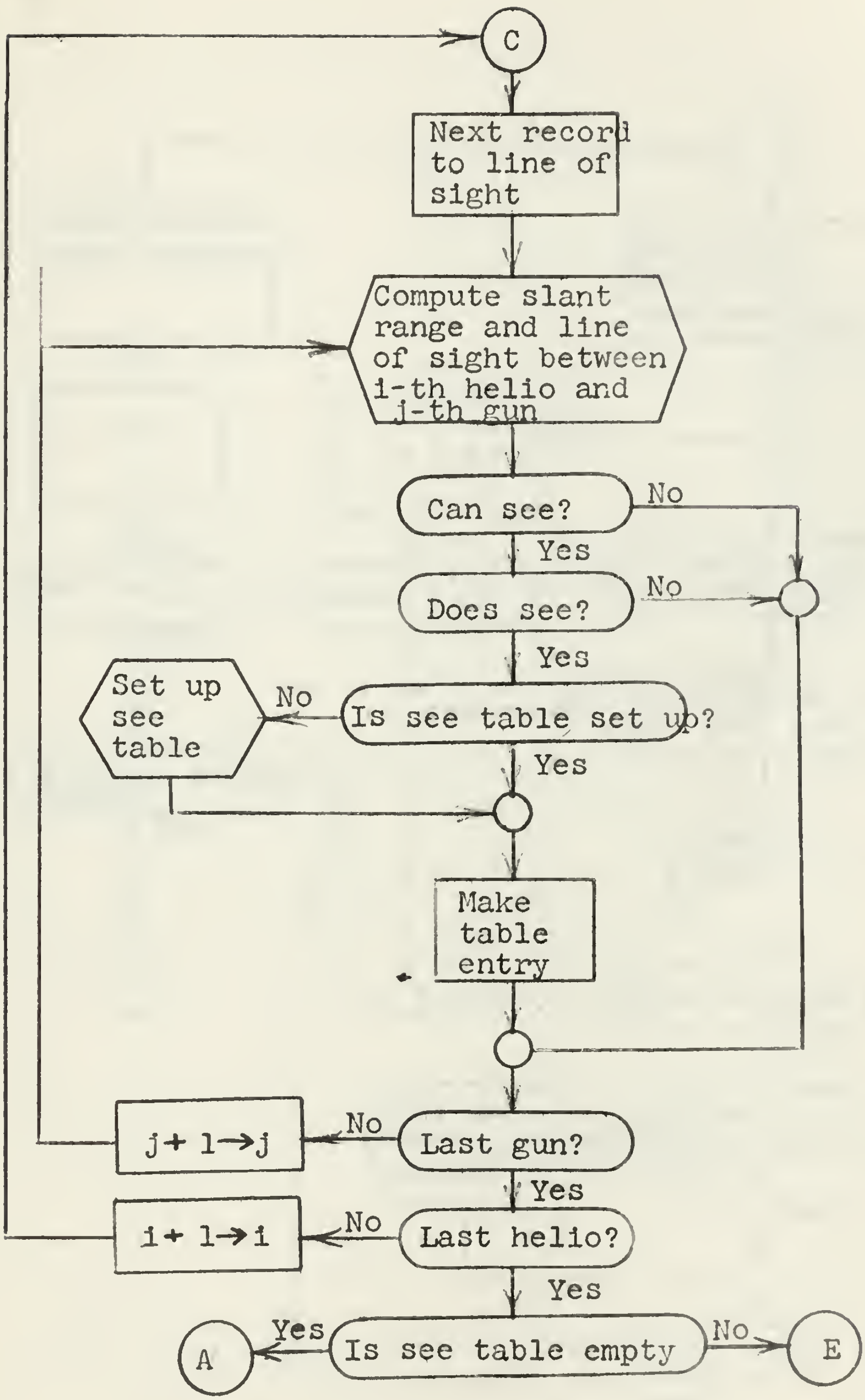



Start and Launch Sub-Routine

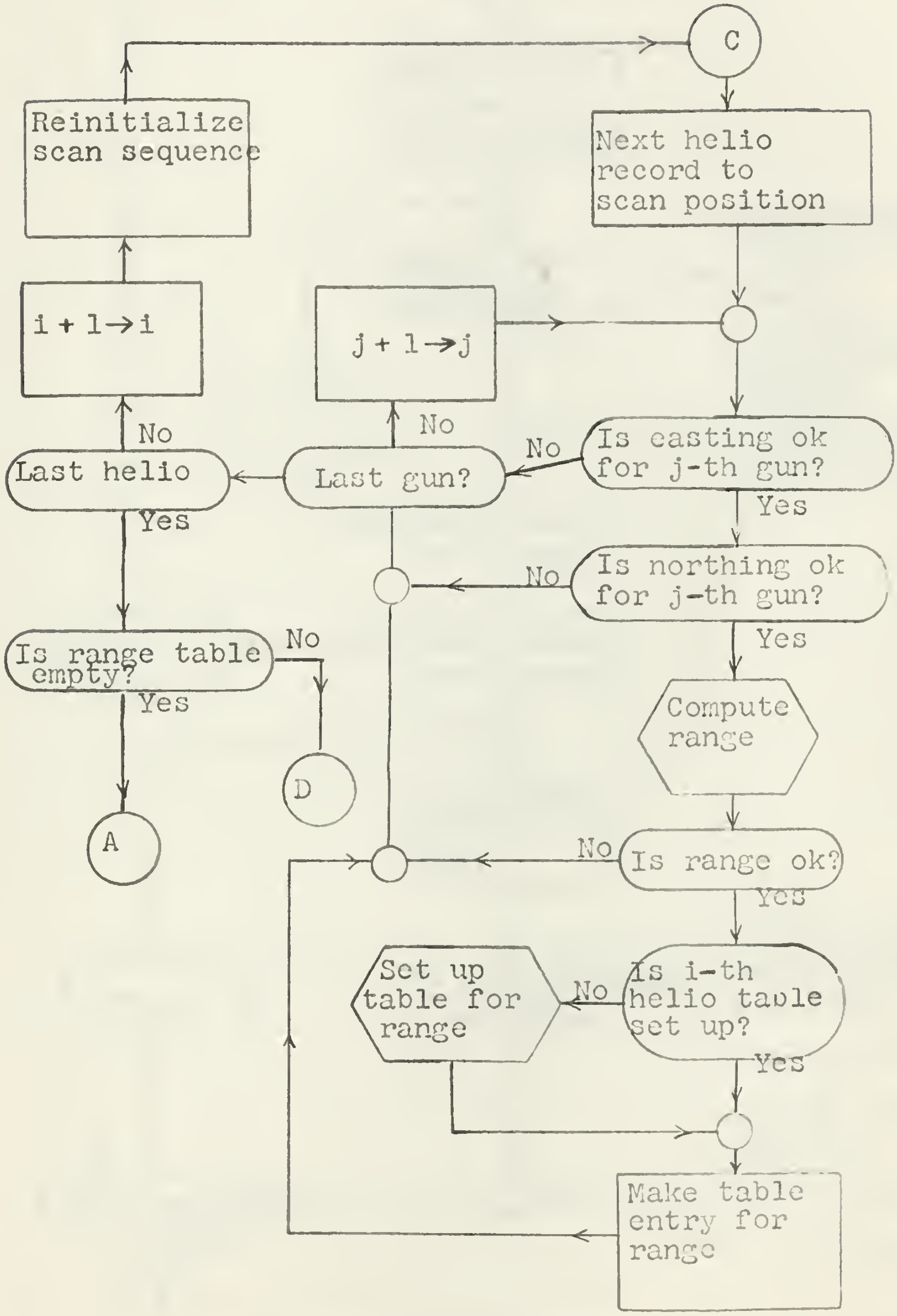
Fig. 5



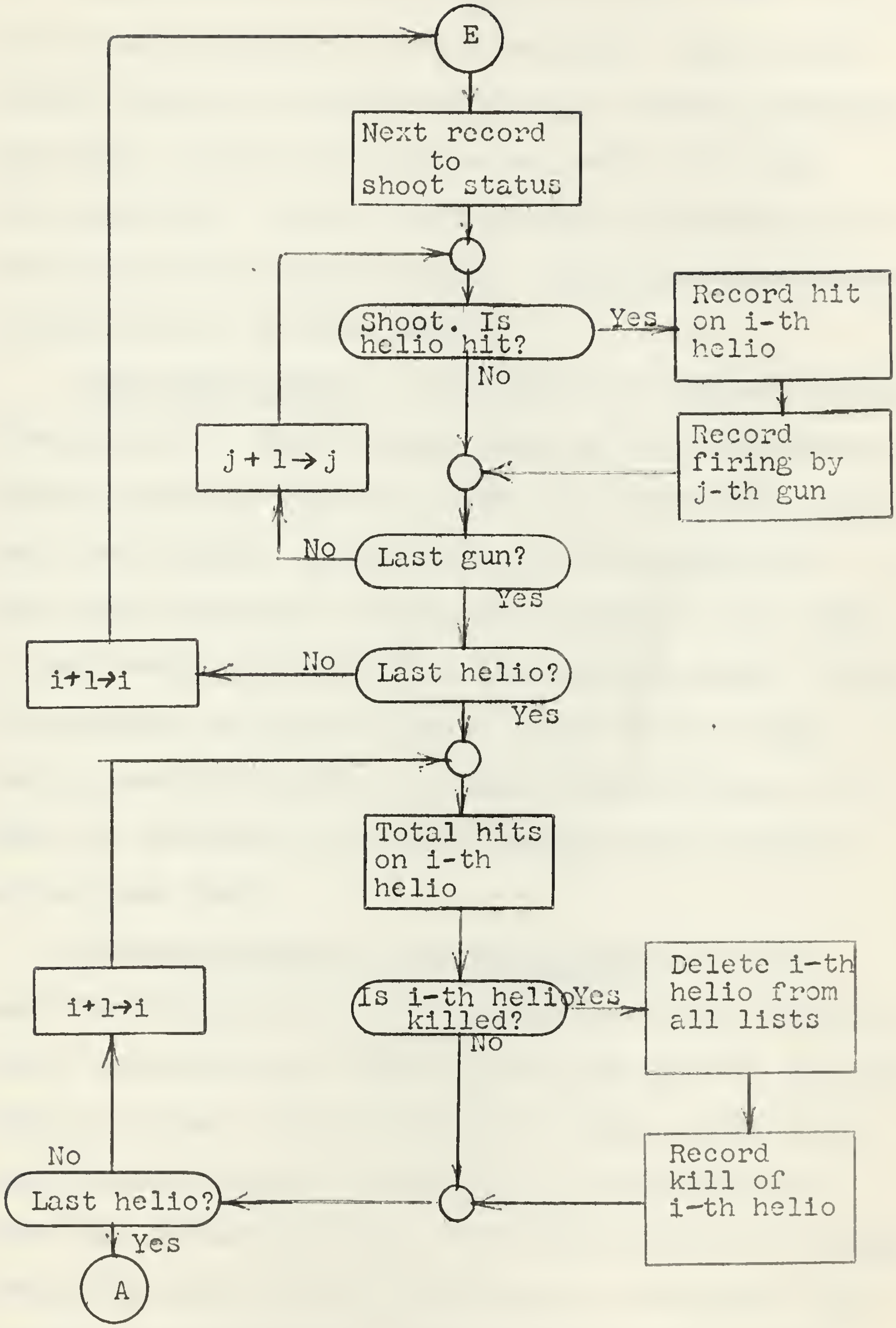

"Game Over"    Sub-routine
Fig. 6




"Can See" Sub-routine
Fig. 7




"Range" Sub-Routine
Fig. 8




"Shoot-Kill" Sub-Routine

Fig. 9

Extensions.

Several areas of this simplified game lend themselves particularly well to extensions. Such extensions, inserted as sub-routines into the main program, will make the game not only more useful, but also more realistic. Some of the possible extensions will be discussed in general terms. Paragraph titles refer to portions of the flow charts.

Helicopter Tracks. To correct the present straight through North - South flight paths of the helicopters, certain types of evasive action could be incorporated into the program. A sub-routine could be written that would allow the helicopters to move to the right or left when they have been fired at and missed. Since the movement of the helicopter right or left might easily make him visible to another gun position, it might be advisable for him to decrease his altitude at the same time.

Offensive Action by the Helicopters. In this game the helicopters are defenseless, but sub-routines could be written that would allow them to fire on-board weapons at guns that have fired at them. Probability distribution functions would have to be set up to cover the probability that they would correctly identify the firing guns and their positions. The shoot, hit, and kill probabilities of the on-board weapons would

also have to be provided.

This offensive action would not have to be limited to on-board weapons. The helicopters could call for supporting fire from either friendly artillery or from friendly aircraft. Either case would require the insertion of a long string of probability functions reflecting the difficulties of identification, location, communications, firing, hitting, and killing.

Move All Pieces. Probably the most difficult extension, and at the same time the most useful one, would be the one that would allow the anti-aircraft weapons to move. Once this has been accomplished, the war game can then be very easily adapted to many more situations much beyond the present scope. Since these weapons would move over some special type of terrain, decision rules must be laid down for their moves. These would have to be of the type discussed earlier in Chapter II (see page 35 ). To accomplish this, the terrain would have to be described in detail, much as a map describes the terrain. A word in storage could be packed with numbers; the locations and values of the numbers, so packed, would indicate the exact nature of the terrain and its natural and man-made features. One word could be used to describe the terrain within each of the grid squares, which results in another list of 10,000 words. This list could be

made up and stored in the computer's memory unit in a manner similar to that for the altitudes of the terrain features previously described. If these terrain quantifiers were stored at addresses that differed only by a constant from that of the grid square they describe, access would be simple and rapid.

An Example of Terrain Quantifiers. The terrain quantifiers for each grid square can be stored in a memory word as code numbers in specific portions of the word. The location and the value of the digit would each have a definite meaning, in accordance with a predetermined code, as follows:

storage word*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | F | E | D | C | B | A |

The two digit** position labeled "A" could represent the topographic quality of the terrain in accordance with the following code:

1 - topographic crest.

2 - military crest.

3 - valley floor.

4 - midway up the slope.

5 - 1/4 up the slope.

6 - 3/4 up the slope.

* The CDC 1604 uses a 48 bit storage word, i.e. 16 octal digits.

** Two digit positions are necessary to represent numbers greater than seven.

The two digit position labeled "B" could represent the trafficability of the terrain in accordance with the following code:

1 - highway.

2 - road.

3 - unimproved road.

4 - trail.

5 - impassable for tanks.

6 - impassable for vehicles.

7 - impassable for personnel.

Numbers in position labeled "C" could indicate the type of vegetation:

1 - dense woods.

2 - woods.

3 - woods with thick underbrush.

4 - bushes or shrubs, 5' or over.

5 - bushes or shrubs, less than 5'.

6 - high grass.

7 - low grass.

Numbers in position labeled "D" could indicate the type of water cover:

1 - fordable stream or river.

2 - unfordable stream or river.

3 - passable swamp.

4 - unpassable swamp.

5 - ocean.

6 - beach.

Numbers in position labeled "E" could indicate the occupation:

1 - unoccupied.

2 - friendly.

3 - enemy occupied.

Numbers in position labeled "F" could indicate the type of field fortification:

1 - foxholes.

2 - trenches.

3 - bunkers.

4 - concrete emplacements.

These terrain quantifiers will enable many complicated decision rules to be employed. When they are stored at addresses differing by only a constant from the address of the grid square that they describe, it is only necessary to add that constant to the address of the grid square to find the address of the quantifier. A suitable mask would be employed to screen out all of the data that is not required, and the information needed to make the decision would be available.

<u>Total Hits on i-th Helicopter</u>. An obvious extension at this place in the program is to require more than one hit by the firing weapon before a kill is made. The program totals the number of hits made upon the helicopter. Several different probability

functions could be provided, which would reflect the increasing probability of a kill with the increasing number of hits.

Shoot. Is Helicopter Hit? At this point in the program a sub-routine could be written that would permit special handling of the firing sequence. If the anti-aircraft weapon is a missile, an exit is made to a sub-routine. This sub-routine carries forward the i-th helicopter and the j-th missile battery to determine if the helicopter is masked during the flight of the missile. If masking exceeds the critical threshold time, no hit is made. If a kill is made, an indicator is set and a return is made to the point of previous exit from the main program. The helicopter is then allowed to fly on until the time of its kill. This allows the other missile batteries to engage it during this time. The missile battery making the kill is suppressed during this time and for some further period, corresponding to its reload time, since it will be out of action for other targets during this time.

Self-Analysis. A sub-routine can be written that will compute the sample mean, sample variance, and any desired statistical test of these computed values. This can be done for each of the outputs of the game. The sub-routine could also include the necessary

commands to cause all of these results to be printed out on any of the available peripheral equipment (electric typewriters, paper tape, magnetic tape, etc.).

## APPENDIX A

## MONTE CARLO METHODS

A great deal of emphasis is currently being placed on an assortment of computational devices and estimation techniques which go under the collective title of "the Monte Carlo technique". The emphasis exists in a number of fields, one of which is war gaming. Another is, surprisingly enough, nuclear physics. This appendix will briefly describe three applications of the Monte Carlo technique, one in nuclear physics, one in the field of numerical analysis, and the last will be the restatement of the numerical analysis application in a war gaming context.

At the outset, one fact should be very clearly understood. There is no one scheme which can be singled out with the remark "THIS is THE Monte Carlo technique". Monte Carlo is the title given to a truly bewildering array of schemes whose common denominator is simply that all are rooted in statistics. Essentially, a probalistic model is built to represent some real or theoretical problem and, by means of artificial generation of random samples, estimates are made of some of the parameters of the system. The Monte Carlo technique consists, not in a particular method of computation, not in a particular method of taking samples, nor in

solution of problems of a particular type. Monte Carlo consists of taking problems which are difficult or impossible of solution by conventional methods, and recasting such problems in terms of a problem in statistics. Although this recasting does not provide a solution of the original problem in the analytic sense, it does allow estimates of the solutions to be made.

A prime example of this estimation process is afforded by reference to one of the first problems to be attempted by the Monte Carlo technique. This problem is that of neutron penetration of shielding in a reactor. The continuing investigation of this same problem accounts for a very large proportion of current Monte Carlo applications. Basically the problem is as follows: nuclear reactors generate a high neutron flux, which can be quite dangerous to human life over a wide area. Therefore, reactors require shielding — shielding which must be guaranteed to reduce the neutron flux through the outermost wall of the reactor to some safe value. That is, a neutron will completely penetrate the shielding with at most probability k.

The calculations required to determine the necessary shielding are impossible to make formally because of the nature of the neutron. The neutron

carries no charge, and hence is completely unaffected by the Coulomb field surrounding electrons and protons. On a nuclear scale, the effects of gravitation are negligible. Therefore the major factor in determining neutron paths through the shielding is collisions. When water or some other amorphous substance such as concrete is used for shielding, one is unable to appeal to the regularities associated with thin crystaline structures in determining the distribution of impact parameters and scattering angles. The most satisfactory model to suit the problem appears to be that of the <u>random</u> (or drunkard's) <u>walk</u>. In this model, the neutron is assumed to travel for a distance, distributed in some fashion about a mean distance and in an entirely arbitrary direction. At the end of the current "leg", it is scattered in a random direction for another randomly distributed distance. To run this model on a digital computer, one sets the initial conditions so that the neutron starts into the shielding with some initial velocity. Four uniformly distributed random numbers are then generated. Three of these are then converted to direction cosines. Using these direction cosines, the fourth is transformed into a distance. (See Appendix B, on random number generation, for the discussion of generation and of conversion to the desired distribution.) A check is then made of the

neutron's new position. The process is continued until the neutron either reenters the reactor, is slowed to "thermal velocity", or completely penetrates the barrier. A record is kept over a large number of trials of the number of trials and of the number of neutrons which succeed in penetrating the barrier. From these figures an estimate of the probability of barrier penetration is made.

Many variations of this problem exist. The method is used extensively in what amount to curve-fitting problems, where different functions for the distance to the next collision are tried and compared to experimental results in an effort to determine the form and parameters of the nuclear binding force function. Other similar estimation problems encompassing a wide range of situations between subatomic particles are investigated by Monte Carlo.

Another technique in the Monte Carlo family involves the evaluation of integrals. This accounts for a substantial portion of Monte Carlo usage outside of nuclear physics. Assume it is desired to evaluate the integral

$$I = \int_a^b f(x)\,dx .$$

By a suitable transformation it is possible to convert the integral to one of the form

$$\hat{I} = \int_0^1 \hat{f}(x)\, dx$$

where $0 \leqslant \hat{f}(x) \leqslant 1$ for $0 \leqslant x \leqslant 1$. Now let us generate a pair $(x_1, x_2)$ of uniformly distributed numbers on the interval $[0,1]$. With the first member of the pair, $x_1$, compute the value of the function, $\hat{f}(x)$. Compare the function value thus obtained with the number $x_2$. If $x_2 \leqslant \hat{f}(x_1)$, tally both a trial and a success.

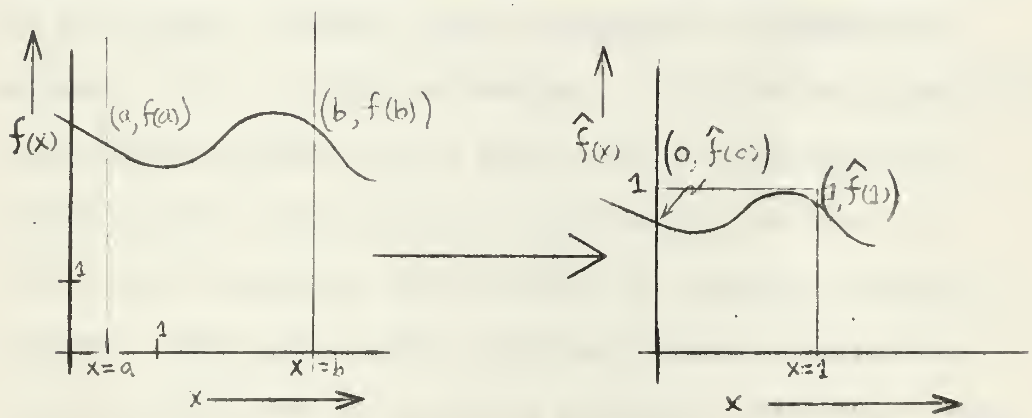



Figure 1.
A graphical representation of the transformation from I to $\hat{I}$.

If $x_2 > \hat{f}(x_1)$, tally only a fair trial. If n is the number of successes in N trials, then the ratio of n to N will, for sufficiently large N, approximate the area of the integral $\hat{I}$, the original integral transformed into the

unit square. Suitable inverse transformation will give the area under the original curve, I.

One may reasonably ask the question, "In what way does this method have an advantage over other numerical methods?" In many relatively simple problems it not only has no advantage over more conventional numerical methods, but frequently suffers badly in comparison. The true advantage comes to light mainly in problems where normal numerical analysis techniques require calculations of nightmarish complexity. An example might be found in application of a technique similar to the Runge - Kutta - Gill numerical integration scheme, say, to five variables. To allow each variable one hundred values, in a step - by - step process would require ten billion iterations, and each of these might require calculation of several function values. The Monte Carlo approach might succeed in giving sufficiently accurate estimates with far fewer calculations.

Let us now examine the reasoning behind the above statements. For a uniformly distributed random variable on the interval $[0,1]$, the probability that the random variable lies in the interval $[a,b]$, $0 \leq a \leq b \leq 1$, is proportional to the length of the interval, $(b-a)$. Thus, if in the example given we hold $x_1$ fixed, and find that $f(x_1) = 0.75$, we will find that 75% of the

uniformly distributed $x_2$ variables are such that $x_2 \leqslant \widehat{f}(x_1)$. Since $x_1$ is not fixed, but is also uniformly distributed on $[0,1]$, we find, for large N, that if the pairs $(x_1, x_2)$ are plotted on the unit square, the distribution of points in the plane is also uniform.

It becomes quite clear, now, that what we have, in essence, done is performed the following operations:

(1) Taken the original curve and compressed it into the unit square.

(2) Assuming the numbers $x_1, x_2$ to have k possible values each, divided the unit square into $k^2$ smaller squares or boxes.

(3) Examined, in random order, a sample of size N of the $k^2$ boxes, determining for each box if the box lies above or below the curve.

(4) From the n successes (n boxes lying below the curve) out of N trials estimated the ratio $\widehat{I}/1 \approx n/N$, which is an approximation of the area under the curve.

In the example war game, an adaption of the procedure used to estimate the area under a curve is employed to simulate the stochastic nature of the outcomes resulting from certain actions taken by the pieces in the game. In the adaptation, the functional values are furnished in the form of a table stored in the computer memory. The circumstances under which the

action is taken determine the particular function value corresponding to the value $\widehat{f}(x_1)$ of the area problem, so that there is no need to generate an $x_1$. However, an $x_2$ is generated and compared to the function value, just as in the area problem. The success or failure criterion is identical, a success being recorded when the function value exceeds the random number generated.

As an example of the adaptation, consider the situation "Tank A fires one round at Tank B, which is one thousand yards away". Associated with Tank A's armament there is a definite probability of a hit on a target located one thousand yards away. This is stored as one of the function values. After the tank-to-tank range has been determined, the hit probability associated with this range is extracted from the table. A random number is generated, and the comparison made between random number and function value. If the function value exceeds the random number, Tank A is credited with a hit on Tank B. Otherwise Tank A is considered to have fired and missed. Although a record is kept of both total shots and total hits made, this information cannot give the area under the hit probability curve since distance to target at first sight is not uniformly distributed. For the tank-to-tank case a reasonable assumption might be that first sight

distance (which is essentially firing distance) is distributed according to the Rolle distribution, with mean equal to the average distance between ridge lines.

The tank-to-tank firing problem above requires that a series of hit probabilities corresponding to different ranges be stored to represent an actual situation. In contrast to this, a single entry might suffice to represent the hit probability curve for an aircraft attacking a ground target. The reasoning is quite simple. Regardless of the distance at which an aircraft first sees a ground target, when the attack is made all bombs are released at approximately the same distance. Hence, for attacks made with high explosive bombs, which are essentially destructive only at the point of impact, only one probability is needed, i.e., the probability that the bomb hits the target. Area destructive weapons such as atomic bombs present a completely different problem due to differences in delivery mode, variation of effect with distance from ground zero, and the wide range of equipment vulnerability.

A practical word about representation of distribution functions and random numbers is now in order. In games played such as the example game, many of the distribution functions are determined empirically.

Experiments are run, data determined and plotted. The accuracy of such data is always open to question. Frequently it is at best an order of magnitude estimate, which may be off by factors of two or three. Therefore the setting up of some complex formula to determine probabilities with more than, say, two digits of accuracy is foolish. Tables should be no more than two- or three- place accuracy. Correspondingly, random numbers may be broken down into blocks of two or three digits, and these used in the comparisons for "above" or "below" the curve. A rather useful by-product of this policy is that, in computers with masking facilities, several distribution functions (or more correctly, probability curves) may be packed into one block of words by assigning curve A to the first three digit positions, curve B to the next three, and so on. However, the main point is that extreme precision generally results in wasted time, and results are no better for it.

Interpretation of results obtained via the Monte Carlo method should be very cautious. For accurate estimates of parameters, sample sizes which can only be described as "fantastically large" must be taken. Where possible, a game with known outcome should be programmed to check on the output of the game. In

scientific computations at least, one reliable answer should be obtained in some other fashion, to determine sample sizes needed to meet the requirements of accuracy.

## APPENDIX B

## PSEUDO-RANDOM NUMBER GENERATORS

John von Neumann has said "Any one who considers an arithmetical method of producing random numbers is, of course, in a state of sin." In the sense of the most rigorous definitions of the word "random" this statement is quite correct, and brooks absolutely no argument. However, there exist a large number of techniques for generating numbers that are termed "sufficiently random", and it is with these so-called "pseudo-random number" generators that we will be concerned. The reason for the use of the term pseudo-random is quite obvious. Since the numbers are of finite length, are discrete, and are generated by a deterministic process, they are not independent in the strictest statistical sense. The arithmetical processes used in generation seldom, if ever, produce the exact distribution desired. Hence the numbers generated are correctly called pseudo-random. However, numbers so generated are treated as if they were random in the strictest sense, and for the remainder of this appendix they will be simply termed random numbers.

Before proceeding further with the discussion of random number generators, it is desirable to point out why the numbers are generated, and not brought into the computer from some auxiliary memory device. To

prepare the table of one million random ten-digit decimal numbers published by the RAND Corporation, several men with a small computer and an elaborate electronic device, worked for approximately six months. Utilization of this table requires that the numbers be transferred from a printed page to some device which can furnish them to the computer, as needed. Since magnetic tape is one of the cheapest forms of auxillary memory, as well as one of the fastest, we will assume the table is converted to magnetic tape for transfer to the computer. To load the magnetic tape we must first convert to punched cards. To do this would require at least ten weeks. To further convert the punched cards to magnetic tape would require another week. To read into the computer memory a block of ten numbers from magnetic tape requires approximately ten milliseconds. Thus, each number drawn from the table requires an average of one millisecond of computer time plus a considerable expediture of time, money and effort prior to going to the computer. To draw the 20,000 random numbers needed to generate the 100 x 100 terrain matrix of the example war game requires some twenty seconds. To generate 20,000 random numbers requires only two seconds. That is, even basing the relative time expenditure strictly on having the numbers brought from magnetic tape into memory the time advantage is of the order of

10 to 1 in favor of generation of numbers over drawing the numbers from a table.

The term "sufficiently random" is at best a poorly defined term. It means that the numbers generated have been subjected to some tests for randomness (that is, they have been treated as a random sample from a given distribution), and have in some fashion "passed" the tests. There are at least as many tests as there are random number generators, and for each test every user has his own idea of exactly what constitutes a "passing score". The situation is quite confused, and shows every promise of remaining that way for some time to come. However, in spite of this it is still possible to bring a sort of partial ordering out of chaos, and this is what we shall try to do. We shall discuss representative random number generators, some of the tests used, and present the results of a test on the random number generator used in this game. The subject of passing scores will not be discussed.

## GENERATION OF PSEUDO-RANDOM NUMBERS.

At the heart of practically every random number generation scheme, regardless of the distribution to be approximated, lies an inner scheme to produce uniformly distributed numbers on the interval $(0,1)$. Given such a set of uniformly distributed numbers, it is

possible to transform them by means of standard techniques of Analysis and/or linear algebra to practically any desired distribution. Linear transformations, shears, magnifications and rotation of axes are the linear algebra techniques most used. From Analysis the principal technique chosen is computation of functional values of particular functions. We shall look briefly at two methods for securing uniformly distributed numbers, and then examine one technique for transforming pairs of uniformly distributed numbers to normal (0,1) numbers via a computation formula from Analysis. These normally distributed numbers will then be transformed via a shear and magnification to numbers distributed normally (m,s). That is, with mean m, and standard deviation s.

Perhaps the most commonly used technique for securing numbers distributed uniformly on the interval (0,1) is the middle-out or mid-square method. To initiate this method, some primitive random number must be chosen as the first number to be squared. This can be done in several ways, but perhaps the easiest is to go to a table such as the RAND table of 1,000,000 random digits and select a sequence of numbers such that, after all eights and nines have been stricken out, a sequence of sixteen octal digits remain. This particular procedure is adapted to the Control Data Corporation 1604 computer, which is the machine locally available. The essential

idea is to choose a number as large as can be fitted into the accumulator of the computer, striking out digits which cannot be fitted into the machine. The 1604 has a sixteen digit accumulator, and does arithmetic in binary with octal mode representation. Hence we choose sixteen digits, and strike out eights and nines. The number chosen is now squared, producing a double length (thirty-two digit) product. The middle sixteen digits of this double length product are extracted, furnishing at once both the next number to be squared, and the random number desired. Symbolically,

$X_{i+1}$ = Middle sixteen digits of $(X_i)^2$

Numbers so generated approximate the uniform distribution quite well as regards frequency of occurrence of digits in each digit position. Results of a test on this generator are shown in Table II.

Another common generator for numbers distributed uniformly $(0,1)$ is an additive scheme accomplished as follows. A table of k random numbers is obtained as described above to serve as primitive numbers and stored in the computer memory. With i in the set of integers modulo k, we perform the addition

$$X_i = (X_{i-1} + X_{i+1}).$$

Overflows are neglected. $X_i$ is then stored in the i-th position of the list of primitive random numbers, and also presented as the random number generated. The most

significant digits of this sequence of numbers approximate the uniform distribution quite well, but the uniform property is definitely lost in the least significant digit positions.

The above processes describe simple techniques for generation of uniformly distributed random numbers. The mid-square technique consumes less time and produces numbers that throughout their whole length are a good fit to the uniform distribution, and thus can be chopped up into shorter numbers so that one iteration can be expected to produce say, four, four-digit uniformly distributed numbers. The best that the additive process can do is produce three four-digit numbers. Therefore, the mid-square generator was chosen for use in the example game. Let us now examine very briefly a procedure for transforming pairs of uniformly distributed numbers into normally distributed numbers with mean zero and standard deviation one. Given the uniformly distributed pair, $X_1$ and $X_2$, we use them in the generation of U:

$$U = (-2 \ln X_1)^{\frac{1}{2}} (\cos 2\pi X_2).$$

U will be normal (0,1). Normality comes from the generation of the logarithm, and the cosine is a scaling factor. It should be noted that most generators, such as the mid-square, can generate zero as a random number, and due caution must be taken to ensure that in this case

the computer stops . Otherwise highly inaccurate conclusions will be drawn from a game run. Before discussing how the normal (0,1) is converted to a normal (m,s), a word about the generation of both natural log and cosine is in order. Several methods are currently used to generate each of these functions, usually by polynomial approximation or series summation. The difficulty lies in the fact that the number of terms needed varies, and with usual methods of nesting polynomials it is necessary to start the computation anew if the chosen number of terms is insufficient. A somewhat different method can be used in lieu of polynomial approximation, with extremely good results and, generally speaking, a minimum of time. This method is that of approximation by continued fractions. The specific continued fraction, or type of continued fraction, is the so-called continued fraction of Gauss, derived from the hypergeometric series. This gives as the expansion for log of $(1+z)$ the fraction

$$\log(1+z) = \cfrac{z}{1 + \cfrac{1^2 z}{2 + \cfrac{1^2 z}{3 + \cfrac{2^2 z}{4 + \cfrac{2^2 z}{5 + \cfrac{3^2 z}{6 + \cdots}}}}}}$$

which is valid for all z greater than -1. Convergence is quite rapid, and if the upper bound on z is known, the iterative loop can be set just once to compute for all values of z. The advantage lies in the number of iterations required.

To compute the cosine of z, we first compute the tangent of z/2 by the formula

$$\tan(z/2) = \cfrac{z/2}{1 - \cfrac{(z/2)^2}{3 - \cfrac{(z/2)^2}{5 - \cfrac{(z/2)^2}{7 - \cfrac{(z/2)^2}{9 - \cfrac{(z/2)^2}{\ddots}}}}}}$$

Then, from the relationship

$$\cos z = \left[1 - \tan^2(z/2)\right] \Big/ \left[1 + \tan^2(z/2)\right]$$

we compute the cosine.

To go from a normal (0,1) distribution to a normal (m,s) distribution we employ a linear transformation involving a magnification (multiply the deviate U by s) and a shear (add m to the product sU). Symbolically,

$$W = m + sU.$$

The sequence of numbers, $W_i$, will be from a normal (m,s) population.

For a more complete discussion of random number generation, the reader is referred to the articles cited in the references.

TESTS FOR RANDOMNESS

We have seen above a few methods for generation of randomly distributed numbers deemed to be "sufficiently random". The question now arises, "What tests have the numbers passed?", and also "How well were the tests passed?". Again we shall look briefly at only a few of the more common tests. "Passing scores" will not be discussed, except to say that the optimum course of action here is to hire the best statistician, and discuss with him fitting the desired distribution, and what, in this particular case, constitutes a good fit.

The most commonly used tests for randomness lean heavily on the Chi-squared test for goodness of fit from the field of statistics. Simply stated, the Chi-squared test is as follows: Consider a sample of N objects that may be sorted into n categories. The probability that an object selected at random will belong in the ith category is $p_i$. Then the number, $e_i$, of objects we expect to find in the i-th category in a sample of size N is given by

$$e_i = Np_i$$

If we take as the number of objects in a sample of size N observed in category i to be $o_i$, then the calculation is

$$\chi^2_{n-1} = \sum_{i=1}^{i=n} \frac{(o_i - e_i)^2}{e_i}$$

With this value we may then enter a table of the Chi-squared distribution, such as is found on page 245 of the Chemical Rubber Company Handbook of Mathematical Tables and, corresponding to n-1 degrees of freedom find P, the probability that a random sample drawn from a population distributed according to the hypothesized distribution will give no better fit (i.e., give no lower Chi-squared value). For example, if we compute a chi-squared value of 4.67 and have 7 degrees of freedom, the table indicates that with probability 0.70 a random sample drawn from the hypothesized distribution will give no better fit. Whether, in the particular case at hand, this a "good" value or not is a question best answered by the statistician.

Having described the Chi-squared test, let us now examine some of the ways in which we can categorize the numbers generated.

a. The interval (0,1) can be partitioned into subintervals, and the number of numbers falling into each subinterval counted. With a uniform distribution we should expect an equal number of numbers in each interval.

b. Each digit position can be considered separately, and the frequency of occurrence of each digit in each position can be tallied. Again with a uniform distribution we should expect an equal number of digit occurrences.

c. Rather than test the entire sixteen digit number, we can break it up into blocks of digits, say in pairs. Then we would divide the interval (0,1) into sixty-four subintervals, and fit numbers into these subintervals.

d. Each digit may be checked to see whether it is even or odd, such a test being a variation of (b) above.

e. The digits may be grouped into blocks of length k, and the blocks analysed as "poker hands". For example, if k is five and we are using an octal mode computer, there are 32768 possible poker hands. A theoretical distribution of poker hands is as shown below in Table I.

| HAND | Number of COMBINATIONS | | PROBABILITY |
|---|---|---|---|
| Bust | 8x7x6x5x4 | 6720 | 0.205078 |
| One Pair | 8x7x6x5x5x2 | 16800 | 0.512695 |
| Two Pair | 8x7x6x5x3 | 5040 | 0.153809 |
| Three of a Kind | 8x7x6x5x2 | 3360 | 0.102539 |
| Full House | 8x7x5x2x1 | 560 | 0.017090 |
| Four of a Kind | 8x7x5x1x1 | 280 | 0.008545 |
| Five of a Kind | 8x1x1x1x1 | 8 | 0.000244 |

TABLE I. Showing the number of hands of each category possible in an octal mode computer if numbers are taken five at a time, and the associated probabilities.

The method for deriving the distribution is obvious from an examination of the second column of the table.

Tests other than the Chi-square may, of course, be applied, and one of the more common non-chi-square tests is the serial correlation test. In essence this test measures the correlation between the occurrence of a digit, say 5, in the jth place of a number and the occurrence of other digits in the (j + k)th place. The number k is called the order of the correlation. For correlation between two adjacent digits, we have $r_1$, the serial correlation coefficient of order one, given by

$$r_1 = \frac{\text{cov}\ (u_j, u_{j+1})}{(\text{var}\ u_j)^{1/2}\ \ (\text{var}\ u_{j+1})^{1/2}}$$

For correlation between two digits separated by k-1 digits, we have for $r_k$, the serial correlation coefficient of order k, the formula

$$r_k = \frac{\text{cov}\ (u_j, u_{j+k})}{(\text{var}\ u_j)^{1/2}\ \ (\text{var}\ u_{j+k})^{1/2}}$$

These formulas are the symbolic formulas only . Computational formulas suitable for use on a computer can be obtained from various statistics texts, generally in the sections concerning serial correlation, autocorrelation or analysis of time series.

Many other tests exist, but the above sample will give a sufficient idea of how number generators are tested. For further detailed discussion, see the references listed under Monte Carlo Technique.

The generator used for this game was the middle-out or mid-square technique. The only test applied was the test (b) above, which was applied to the entire 32 digit product. This test indicates that the middle 16 digits do, in fact, give a good approximation to the uniform distribution. The results of the test, on a sample of size 500,000 is shown in Table II on page 106 of this appendix.

In an attempt to discover the mechanism whereby the uniformity of the numbers generated comes about, an interesting piece of information came to light. It was possible to check on the uniformity of the two least significant digits of the 16 digits extracted from the middle of the square by deriving a theoretical non-uniform distribution for the two least significant digits of the square with the assumption that the two least significant digits of the middle 16 are uniformly distributed.

By considering a table of squares of the numbers 00 through 77 the frequency of occurrence of the digits 0 through 7 in the last and the next to last positions is as follows:

| Digit | Probability of occurence in least significant digit position | Next most significant digit position |
|---|---|---|
| 0 | 4/16=.25 | 5/16=.3125 |
| 1 | 8/16=.50 | 1/16=.0625 |
| 2 | 0 = .0 | 3/16=.1875 |
| 3 | 0 = .0 | 1/16=.0625 |
| 4 | 4/16 = .25 | 3/16=.1875 |
| 5 | 0 = .0 | 1/16=.0625 |
| 6 | 0 = .0 | 1/16=.0625 |
| 7 | 0 = .0 | 1/16=.0625 |

The observed probabilities of occurrence were

| Digit | Probability of Occurrence | | | |
|---|---|---|---|---|
| | Last Digit | | Next to last digit | |
| | Predicted | Observed | Predicted | Observed |
| 0 | .2500 | .2500 | .3125 | .3130 |
| 1 | .5000 | .5001 | .0625 | .0627 |
| 2 | 0 | 0 | .1875 | .1862 |
| 3 | 0 | 0 | .0625 | .0621 |
| 4 | .2500 | .2499 | .1875 | .1880 |
| 5 | 0 | 0 | .0625 | .0628 |
| 6 | 0 | 0 | .0625 | .0626 |
| 7 | 0 | 0 | .0625 | .0623 |

Thus analysis of digit positions other than those in the middle of the square provides an additional indication of uniformity in the distribution of random numbers produced. The above analysis can be extended to the last significant 8 digits of the middle 16 of the square with the expenditure of some six hours of computation to compile theoretical distributions, but further extension consumes extensive machine time. Another item resulting from analysis of the entire 32 digit number concerns the relationship between the uniformity and the length of the numbers squared. It appears that of the factors tending to produce uniformity,

the most important ones are the summation process and the cascading of carries. The greater the number of digits in the numbers to be squared, the more uniformly distributed are the digits resulting from both the summation process and the carries. While long word length is not a necessity, it appears to be a distinct aid to securing a more nearly uniform distribution with the mid-square technique. It suggests that for work where distributions very close to the uniform are desired, even multiple precision arithmetic might be resorted to in an attempt to improve on mid-square methods in computers of short word length.

It should be emphasized that the preceding comments are based solely upon the chi-squared test for goodness of fit. Serial correlation and poker tests might well indicate that the apparent advantage of long word length, either computer construction or multiple precision arithemetic derived, may not of itself genuinely aid in construction of uniform, uncorrelated random numbers.

Observed Frequency of Occurrence in Digit Position

| Digit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0 | 62517 | 62137 | 62614 | 62730 | 62291 | 62752 | 62527 | 62911 |
| 1 | 62268 | 62917 | 62471 | 61950 | 62736 | 63026 | 62432 | 62457 |
| 2 | 62925 | 62356 | 62386 | 62526 | 62381 | 62364 | 62648 | 62611 |
| 3 | 62580 | 62639 | 62380 | 62238 | 62574 | 62371 | 61940 | 62400 |
| 4 | 62510 | 62547 | 62614 | 62895 | 62499 | 62374 | 62558 | 62391 |
| 5 | 62320 | 62600 | 63030 | 62346 | 62299 | 62683 | 62687 | 62174 |
| 6 | 62525 | 62318 | 61935 | 62289 | 62716 | 62336 | 62741 | 62452 |
| 7 | 62355 | 62486 | 62570 | 63026 | 62504 | 62094 | 62467 | 62604 |
| $\chi^2_7$ | 4.725 | 6.260 | 10.548 | 14.810 | 3.297 | 9.863 | 7.014 | 5.190 |

Observed Frequency of Occurrence in Digit Position

| Digit | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 62431 | 62630 | 62281 | 62145 | 62543 | 61947 | 62516 | 62660 |
| 1 | 62811 | 63036 | 62554 | 62557 | 62388 | 62138 | 62671 | 62328 |
| 2 | 62225 | 62322 | 62732 | 62459 | 63028 | 62479 | 62339 | 62635 |
| 3 | 62762 | 62317 | 62406 | 62654 | 62226 | 63014 | 62602 | 62303 |
| 4 | 62470 | 62390 | 62363 | 62536 | 62565 | 62595 | 62783 | 62583 |
| 5 | 62417 | 62655 | 62176 | 62518 | 62590 | 62666 | 62270 | 62178 |
| 6 | 62499 | 62222 | 62691 | 62691 | 62469 | 62597 | 62811 | 62704 |
| 7 | 62385 | 62428 | 62697 | 62440 | 62281 | 62564 | 62008 | 62609 |
| $\chi^2_7$ | 4.268 | 7.807 | 5.772 | 3.142 | 6.878 | 12.025 | 8.602 | 4.421 |

Table II

Expected Frequency of Occurrence = 62,500

Sample Size = 500,000

Digit Positions Numbered from Left to Right

## APPENDIX C

## DIGITAL COMPUTER SUITABILITY FOR WAR GAMING

Although the primary concern of this paper is war gaming, per se, it would be incomplete without some reference to the computation equipment. The available equipment has a direct bearing on model design and complexity, and failure to consider the computer in planning a game can lead only to disaster. In this appendix it is intended to discuss computer descriptions in general, and to point out some specific and critical features of computers that greatly affect gaming possibilities.

The description of a digital computer can take many forms, varying greatly in length, in degree of technical information presented, in clarity, and in usefulness. The ideal is for the gamer to be able to read and to absorb all possible technical information about the computer, and to know its minutest detail so that full advantage may be taken of all pertinent features. This is not generally possible, and the gamer must usually be satisfied with a short, reasonably non-technical description of the machine, together with a knowledge of the repertoire of instructions available. It can safely be assumed that, having chosen a machine to employ, the gamer will learn the instruction repertoire in order to design the game. The crucial questions are then:

(1) with a given computer available, is it feasible even to try a game on it?; and (2) given some freedom of choice, what computer should be used? These questions must be answered by consideration of a computer description supplemented, of course, by a more detailed inquiry. Let us therefore discuss briefly the subject of computer description, give an example of a description using the Control Data Corporation 1604 Computer, and then mention certain highly desirable characteristics.

The descriptors for a computer fall generally into six categories which are:

1. Construction and mode of operation.
2. Speed of computation.
3. Memory description.
4. Input-output features and equipment.
5. Instruction and word format.
6. Special features.

To be perfectly honest, a seventh category of description must be added--cost. However, this has no bearing on the suitability of a computer for war gaming, only on its attainability, and is a matter to be settled on an individual basis between the gamer and his comptroller.

1. Construction and mode of operation. Construction generally refers to the electronic components comprising the operating portion of the computer. In digital computers this means either vacuum tubes or transistors. Transistorized computers, which tend to operate trouble-free for longer periods of time, are said to be solid-

state computers. Modes of operation are either serial or parallel.

A serial mode computer is a machine which performs the requisite operations serially down the length of the word. A magnetic drum machine is a good example of a serial mode computer in that it "peels" the word off the drum a digit at a time, and, when possible, operates on each digit as it comes off the drum. Parallel mode machines, on the other hand, operate down the entire length of a word simultaneously. Machines with core memories are generally of this type. If carries or overflows occur as a result of an operation, they are sometimes made by a second simultaneous operation down the whole word, which is repeated until no further carry flags exist as a result of previous operations.

At worst a parallel machine may be almost, but not quite, as slow as a serial machine. Generally, parallel mode computers are one or more orders of magnitude faster in operation than serial mode computers.

2. Speed of computation. This refers to the average time required to perform one addition, and is expressed in fractions of a second per operation, or as operations per second. Alternately it may refer to basic multiply time, or to average time to bring to the accumulator one word from memory, but the most usual reference is to add time.

3. Memory description. This refers to storage media, word size, and format and number representation. Storage media may be cores, a drum or drums, electrostatic storage units, mercury delay lines, disk storage units, or a series of flip-flop switches. Current trend is to magnetic cores, as they represent one of the fastest, most stable non-volatile types of storage media. Word size is expressed as the number of alpha-numeric characters, decimal digits or bits comprising a word. Generally, word length is fixed for binary machines, but both alpha-numeric and decimal machines may have variable word lengths. Of the three types, binary is the most flexible in that it may be treated as both decimal and alpha-numeric for many purposes, whereas the other types may not be treated as binary. Word format has two subdivisions, referring to interpretation of a word as a number (constant) or referring to interpretation of the word as an instruction. The numerical interpretation is concerned with the number of digits represented, and the method of indicating positive and negative numbers. The instructional interpretation is concerned with the number of instructions per word and the number of address referrals per word. Number representation is concerned with the base of the number system. Number representations in use currently include binary, octal, decimal, and hexadecimal numbers*. This

*Referring respectively to base 2, 8, 10 and 16.

is somewhat misleading in that, except for a very few machines, all arithmetic done is binary format, but console and input-output representations are made by grouping bits in order to reduce interpretive work and to speed transfer of information into and out of the machine.

4. Input-output features and equipment. This concerns means and equipment available for getting information into and out of the computer. The fastest means is direct transfer to and from an auxiliary core memory device to a core memory, but cost and complexity severely limit the use of core units as auxiliary memory devices. Next in order of transfer rates, and first in order of capacity, are magnetic tape units. These represent the best compromise between speed, capacity and cost, and banks of multiple tape units are available with virtually every computer built today. One reservation must be made concerning magnetic tape speeds. Information is stored serially on magnetic tapes. Consequently, the ordering of records on the magnetic tape must be done carefully, and with much forethought. Otherwise so much time will be lost rewinding tapes and searching for records that magnetic tape will lose the advantage inherent in its high transfer rates. Another device with transfer rates comparable to those of magnetic tape is the disc file. Disc files have random

access to all records stored on the discs. By virtue of this property, disc files can in many cases exceed magnetic tape units in effective transfer rates. The capacity of disc files is of the same order of magnitude as that of magnetic tapes. Magnetic drums are also used as auxiliary memory devices, principally on machines with core memories. Capacity is limited on a drum, but transfer rates to main memory are moderately high. Drums have a degree of random access to all locations on the drum similar to that found in disc files. Drums are excellent both as auxiliary memory devices and as buffering equipment between main memory and the much slower input-output devices, such as typewriters, card punches, line printers and paper tape punches. Card reading and punching machines are probably the next fastest, and certainly one of the most flexible means of input-output. Inasmuch as nearly all input data is prepared on cards at some stage prior to actual loading into the machine, and since one card of a stack may easily be modified, inserted or deleted, it is frequently desirable to have this capability in the computer system, even at the much slower input-output speeds (compared to magnetic tapes). Paper tape readers and punches vie with cards insofar as speed is concerned, but they lack the capability of easy and rapid modification of cards. Typewriters connected directly to input-output

lines are frequently available to make information print-outs and very short entries, but are by far the slowest automated method of input-output. Manual entry switches are a standard feature, but are input devices suitable only for one-cell changes. Console display lights can be used to display single cells or groups of cells, as can cathode ray tube display devices, but these are not really classifiable as output devices. Other special devices such as line printers may be connected, but the above listed devices comprise the usual array available.

5. Instruction and word format. Instruction format refers to actual instruction code size and format, index register designation, and addressing structure. Operation codes may be several digits long, usually two or three, and tell the computer which operation to perform. This is usually considered unimportant in most respects. Index register designators are one or two digits, depending on the number available. Sometimes an alphabetic character is used as in the IBM 650. Addressing structure may include one, two, three, or occasionally four addresses per word. These addresses refer to the locations of operands or to the address of the next instruction. Occasionally the address itself may be an operand. Word format was discussed under Memory description.

6. Special features. This may be used to cover a multitude of items, which are not quite common enough to be accepted as standard. As designs advance, these features become standard and are then mentioned only if they are not available. Items in this category change quite rapidly. Currently, a typical list might include items such as indirect addressing, floating point arithmetic, special input-output devices, some logical-type operations and new or unusual commands.

As an example of how these descriptors are used, a description of the Control Data Corporation 1604 Computer is given as follows:

> "The CDC 1604 is a high-speed solid state digital computer. Operation is binary, parallel mode. Memory consists of 32,768 words of core storage. Word length is 48 bits, with two single address instructions per word. Number representation is in octal format, giving 16 octal digits per word. Basic add time is 4.8 microseconds. Input-output equipment includes a high-speed Ferranti paper tape reader, paper tape punch, console typewriter, manual entry switches, console display, four magnetic tape units with read-write capability of 30,000 characters per second, and high-speed direct linkages with other computers in the CDC family. Special features include 6 five-digit index registers, floating point arithmetic with exponent range of two to the plus or minus 1023, bit-wise logical operations and indirect addressing."

While this description is by itself insufficient to furnish the basis for a decision as to whether or not to request a particular computer for gaming purposes, if it is taken together with a detailed command list and a table of instruction execution times, it can very

adequately furnish the needed basis for decision on the part of the gamer. Other more technical considerations may cause rejection of the computer by other personnel of the gaming group.

Of the features pointed out in the discussion, which ones should the war gamer consider as the most critical, and why? In the final analysis, the critical features are those which determine the possibility of ever doing the game on the computer (i.e., Can it be done in some reasonable fashion?), and which determine the amount of time which need be taken. Few features can be listed which contribute solely to determining one or the other of these two pieces of information. For example, indexing in a machine which lacks this feature may usually be accomplished by means of a pseudo-operation, but the time penalty to be paid may be so great that indexing is "impossible." If the game absolutely requires indexing, the gamer must then seek another machine to run the game. Of course some features, such as memory capacity and basic speed, cannot be programmed around, and a lack in either of these two categories furnishes a quick and easy decision. Some critical features are listed in the following paragraphs.

The three most critical features of a machine for war gaming are the availability of binary format, logical operations on bits and long word length. The

reasons therefor are not quite obvious and require some explanation. In virtually every type of war game, extensive tables indicating the status of elements of the game are mandatory. For example, if we have an aircraft involved, we need to know its location, speed, direction of flight, offensive and defensive status, origin, destination, payload and designator. With one item per word the above list takes a minimum of 13 words if location and direction of flight are taken to include three items each. For a piece of terrain we might want to know its coordinates, altitude, occupancy status, vegetation type, type of man-made objects on it, whether it is water or land, and if it is water, is it a lake or a stream, and if a stream, direction of flow and fordability, and so on. At the rate of one word per item and for a piece of terrain gridded with 100 grids per side it is obvious that no computer in existence could even hope to contain the information. Hence it is mandatory in a large scale problem to be able to pack several items into a single word and call a specific item in at will. In this manner a telescoping by as much as a factor of ten can be made. For example, with four bits we can answer the following four questions:

1. Is grid square XY occupied?
2. Is the occupant a friend or foe?
3. Is the occupant a tank or an infantry unit?
4. Is the tank a heavy or medium? Or is the infantry unit armed with bazookas?

In a 48-bit per word machine, this telescoping operation compresses four words to two percent of the space occupied by one word per piece of information. In a complex game this telescoping is vital, and the inability to do it is, as it were, fatal. Thus, the need for logical operations on bits, which will permit this packing is established as crucial. Long word length has a value which is obvious from the previous discussion. With much information needed and instantly available, long word length and word packing becomes a critical factor. Otherwise the time consumed by input-output operations mounts to an unacceptable degree. The remaining one of the three factors concerns simply the need for binary representation within a word. In a binary machine logical operations may be easily accomplished because one can generate a mask of any number of bits in length. Any one bit or group of bits may be extracted and examined. In a decimal machine, it is usually impossible (to generate a mask which is guaranteed) to bring in all bits in a group of four for examination. The alternative is to not use the bit structure to the fullest, or to pay a heavy penalty in time spent dumping digits off the ends of a register in order to isolate one digit. Binary representation gives an economy of usage which it is <u>extremely</u> difficult

to bypass, and should be considered as a critcal factor in considering a machine for use.

The next most critical grouping of factors again concerns the computer memory. It is the size and speed of access to both the main memory and auxilitary memory units. Except in the case where main memory is a drum and auxiliary memory is a magnetic tape unit or core bank, the access to main memory is at least an order of magnitude faster than access to auxiliary memory. If constant reference must be made to an auxiliary memory device to compensate for small main memory size, running time for a game can be increased ten-or even a hundred-fold. Therefore the possession of a relatively large memory and fast means to dump in big blocks of input data loom very large in determining running time and hence, indirectly, even the possibility of ever running the game. The ideal combination appears to be a large core memory (32,768 or more words)* and a bank of several magnetic tape units (8 or more tape units).

*It should be noted that in most binary machines the number of words available as memory is usually a power of two. For example, 32,768 is the fifteenth power of two. This is done to accommodate the largest number of memory cells available for a given possible address length, and for technical reasons such as facilitating "end-around" addressing systems where cell 00000 is the cell following cell 77777.

Another important item to consider is speed of operation. This bears, via the time element, on the possibility of doing the game at all. In general, computers may be grouped under three headings as regards speed. High speed computers need on the order of ten micro-seconds per operation. Intermediate speed computers take approximately five hundred microseconds per operation. Slow computers require several milliseconds per operation. A week of computation on a high speed computer becomes a year of computation on an intermediate speed machine and perhaps a decade of computation on a slow machine. Stated another way, the gamer has a choice between a high speed computer or a fairly simple (and therefore usually highly unrealistic) game.

APPENDIX D

BIBLIOGRAPHY


1. Ablow, C.M., and Brighman, G., "An Analog Solution of Programming Problems", JORSA, Vol. 3, page 388-394.

2. Adams, E.H., and Forester, R.E., "Carmonette: A Computer Combat Situation", ORO/JHU, Nov. 1959.

3. Andlinger, G.R., "Business Games- Play One", Harvard Business Review, March - April 1958, page 115-125.

4. Antosiewicz, H.A., "An Analytic Study of War Games", NLRQ, Vol. 2, No. 3, Sept 1955, page 181.

5. Bellman, Clark, Malcom, Croft, and Ricciardi, "On the Construction of a Multi-Stage Multi-Person Business Game", JORSA, Vol.5, 1957, page 469-503.

6. Berkowitz, B.J.,Christiansen, J.G.,Gillie, J.B., Girard, E.W., and Ryan, R.B., "A Combat - Intelligence War Game(U)", ORO/JHU, SP-129, April 1960, SECRET.

7. Blackett, D.W., "Some 'Blotto' Games", NLRQ, Vol.1, No.1, March 1954, page 55-60.

8. Blackett, D.W., "Pure Strategy Solutions of 'Blotto' Games", NRLQ, Vol.5, No.2, June 1958, page 107.

9. Brackney, H., "The Dynamics of Military Combat", JORSA, Vol. 7, No.1, page 30.

10. Blackwell, D., "On Multi-Component Attrition Games", NLRQ, Vol.1, No.3, Sept. 1954, page 210-216.

11. Brown, G.W., "Iterative Solutions of Games by Fictitious Play", Cowles Commission Monograph,

12. Carr, H.W., "A Monte Carlo Model of an Air Force Type Supply System", Rand, P-638A, May 1955

13. Claywood, T.E., and Thomas, C.J., "Applications of Game Theory in Fighter versus Bomber Combat", JORSA, Vol.3, 1955, page 402-411.

14. Clark, C.E., and others, "War Gaming, Cosmagon and Zizspeil", The Johns Hopkins University, 1957.

15. Clark, D.K., "War Game Manual for Umpires and Evaluators(U)", ORO/JHU, T-74, Oct. 1957, SECRET.

16. Cushen, W.E., "Generalized Battle Games on a Digital Computer", ORO/JHU, T-263, Feb. 1954.

17. Cushen, W.E., "War Gaming at Ballistics Research Laboratories" ORO/JHU, S-444, Oct. 1954.

18. Cushen, W.E., "War Games and Operational Research", Philosophy of Science, Vol. 22, No.4, Oct. 1955.

19. Danskin, J.M., "Fictitious Play for Continuous Games", NLRQ, Vol. 1, 1954, page 313-320.

20. Deems, P.S., Col. "War Gaming and Exercises", Air University Quarterly Review, Winter 1956-1957, page 98-126.

21. Dooher, M.J., (Ed.), "The Development of Executive Talent", American Management Association, 1953.

22. Dresher, M., "Methods of Solution in Game Theory", Econometrica, Vol. 18, 1950, page 179-181

23. Dresher, M., and Shapley, L.S., "Summary of Rand Research in the Mathematical Theory of Games", Rand, RM-293, July 1949.

24. Dunn, P.F. and Hitchman, N., "A Design for a Hand Played Tactical War Game", ORSA, Meeting, May 1958.

25. Eddy, A.G., "Applications of the Automatic Data Processing System to the Syntac System of War Gaming", CONARC, CORG-R-82, Aug. 1959.

26. Ellsberg, D., "Theory of the Reluctant Duelist", American Economic Review, Vol. 46, 1956, page 909-923.

27. Edmondson, H.P., "The Distribution of Radial Error and its Statistical Applications in War Gaming", JORSA, Vol. 9, Jan 1961, page 8.

28. Everett, H., "Recursive Games", Annals of Mathematics Studies, Vol. 39, page 47-78.

29. Enke, S., "Logistics Laboratory Problem I, After Two (Simulated) Years", Rand, RM-1993, Oct 1957.

30. Fisher, R.A., "Randomization and an Old Enigma of Card Play", Mathematical Gazette, Vol. 18, 1934, page 294-297.

31. Friedman, C., "Weapon Evaluation by Computer Simulation", Computers and Automation, Vol.7, Oct. 1958.

32. Fulkerson, D.R., and Johnson, S.M., "A Tactical Air Game", JORSA, Vol.5, No.5, Oct 1957, page 704-712.

33. Fusca, J.A., "Navy Warfare Simulator Trains Officers", Aviation Week, Vol.69, No.34, Nov. 10, 1958.

34. Gainen, L., Levine, R.A. and McClothlin, W.H., "Baselogs - A Base Logistics Management Game", Rand, RM-2086, Jan 1958.

35. Galler, B.A., and Dwyer, P.S., "Translating the Method of Reduced Matrices to Machines", NLRQ, Vol.4, 1957 page 55-71.

36. Gamow, G., "A Tank Battle Game", JORSA, Vol.1, No.2, Feb. 1953, page 85.

37. Garrison, J.H., "Logex - Practical Experience in Logistics", Military Review, Sept. 1954, page 50-54.

38. Geisler, M.A., "The Simulation of Large Scale Military Activity", Management Sciences, Vol.5, July 1959, Page 359-368.

39. Green, P.E., and Calhoun, S.R., "Operational Gaming Applications in a Steel Company", ORSA Meeting, May 1958.

40. Gunning, W.F., "Aspects of Real Time Simulation", IRE Transactions on Electronic Computers, Vol. EC-7, No. 2, June 1958, page 134-136.

41. Hale, J.K., and Wicke, H.H., "An Application of Game Theory to Special Weapons Evaluation", NLRQ, Vol.4, 1957, page 347-356.

42. Hamburger, W., "Monopalogs, An Inventory Management Game", Rand, RM-1579, Jan. 1956.

43. Haythorn, W.W., "The Uses of Simulation in Estimating Intrasquadron Logistics Requirements", Rand, P-1656, Feb.1959.

44. Haywood, O.G., "Military Decision and Game Theory", JORSA, Vol.2, 1954, page 365-385.

45. Haywood, O.G., "Military Decision and the Mathematical Theory of Games", Air University Quarterly Review, Sept. 1950, page 17-30.

46. Heilberg, E., "War Gaming of Casualty Producing Chemical Weapons(U)", CONARC, CORG-M-94, Jan. 1960, SECRET.

47. Hodgson, J., "Hexpiel II, a Monte Carlo Analogue to Ground Combat Involving Large Units", ORSA Meeting.

48. Hodgson, J., "Military Intelligence in War Gaming", ORSA Meeting.

49. Isaacs, R., "Pursuit Games, ORSA Meeting.

50. Isbell, J.R., and Marlow, W.H., "Attrition Games", NLRQ, Vol.3, No.1 & 2, May-June 1956, page 71-94.

51. Isbell, J.R. and Wagner, F.G., "Military Evaluation and Statistical Decision", Ballistic Research Laboratory Memo. Report No. 1014, June 1956.

52. Jackson, J.R., "UCLA Executive Game No. 2: A Preliminary Report", Management Sciences Research Project, Paper No.66, April 1958.

53. Jacobs, W., "Air Force Progress in Logistics Planning", Management Sciences, Vol.3, 1956-57, page 213-224.

54. Johnson, M.S., MajGen., "War Gaming and Management", Proceedings of the Third Annual Conference on Operation Research Society for Advancement of Management, Feb. 1958.

55. Kahn, H. and Mann, I., "War Gaming", Rand, P-1167, July 1957.

56. Ladd, D.W., and Wolf, E.W., "A Non-Real Time Simulation of Sage Tracking and Bomarc Guidance", IRE Transactions on Electronic Computers, Vol. 8, March 1959, page 36-41.

57. Leibowitz, M.L., and Levine, E., "Strategic Allocation of Forces: A Combined Game Theoretic and Lanchester Equation Model", ORSA Meeting.

58. Luce, R.D. and Raiffa, H., "Games and Decisions", John Wiley and Sons, Inc., 1957.

59. Machol, R.E., "The Mechanical Blackboard", JORSA, Vol.5, page 422-428.

60. Magruder, E.R., "Decision Theory and Objectives Selectivity", unpublished report, Air Force Institute of Technology, 1957.

61. Malcolm, D.G., "Army Battalion Maintenance Simulation", Report of Systems Simulation Symposium, May 1957, page 88-92.

62. Malcolm, D.G., "System Simulation - A Fundamental Tool for Industrial Engineering", Journal of Industrial Engineering, May-June 1958.

63. Marshak, J. and Mickey, M.R., "Optimal Weapon Systems", NLRQ, Vol.1, 1954, page 116-140.

64. McDonald, J. and Ricciardi, F., "The Business Decision Game", Fortune, March 1958, page 140-142, 208.

65. MacDonald, J. and Tukey, J.W., "Colonel Blotto: A Problem of Military Strategy", Fortune, June 1949, page 102.

66. McDonald, J.D., "Strategy in Poker, Business and War", Norton, 1950.

67. McKinsey, J.C.C., "Introduction to the Theory of Games", McGraw - Hill Book Co., Inc., 1952.

68. Milnor, J. and Shapley, L.S., "On Games of Survival", Annals of Mathematics Studies, Vol.3, No.39, 1957.

69. Mood, A.M. and Specht, R.D., "Gaming as a Technique of Analysis", Rand, P-579, Oct 1954.

70. Mood, A.M., "War Gaming as a Technique of Analysis", Rand, P-899, Sept. 1954.

71. Mulligan, J.E., "A Basic Management Tool for War Gaming Efforts", U. S. Naval Weapons Lab, Dahlgren, Va., ORSA Meeting.

72. von Neumann, J., "A Model of General Economic Equilibrium," Review of Economic Studies, Vol. 13, 1945-46, page 1-9.

73. Newcomb, P.R., "The Analysis of War Gaming", ORO/JHU, ORO-640, May 1956.

74. Novic, D., and Fisher, G.H., "The Role of Management Tools in making Military Decisions", Armed Forces Management, Sept-Oct-Nov. 1956.

75. Pearman, J.P.T. and Whitmore, W.F., "Limitations on the Predictive Use of War Gaming Techniques", ORSA Meeting 1954.

76. Perkins, F.M., "Optimum weapon Deployment for Nuclear Attack", JORSA, Vol. 9, Jan. 1961, page 77.

77. Pritscher, A.A.B., van Buskir, R.C. and Wetherbee, T.K., "Simulation to Obtain a Systems Measure of an Air Duel Environment", IRE Transactions on Electronic Computers, Vol. 8, March 1959, page 55-59.

78. Rafferty, J.A., "Military Applications of Game Theory: A Perspective", Operations Analysis Technical Memo. No. 33, H.Q. USAF, April 1952.

79. Rehkop, J., "Experience with the Management - Decision Simulation Game, Monopologs", Rand, RM-1917, July 1957.

80. Ricciardi, F.M., "Business War Games for Executives, A New Concept in Management Training", Management Review, May 1957, page 45-56.

81. Ricciardi, F. and others, "Top Management Decision Simulation: The A M A Approach", American Management Assn. Inc. 1957.

82. Robbins, J.J., "A Tank Duel with Game Theory Implications", NLRQ , Vol. 4, No.2, June 1957, page 131.

83. Rich, R.P., "Simulation as an Aid to Model Building", JORSA, Vol. 3, Feb. 1955, page 15-19.

84. Robinson, J., "An Iterative Method for Solving a Game", Annals of Mathematics, Vol. 54, 1951, Page 296-301

85. Rowan, T.C., "Simulation in Air Force Systems Training", Report ofSystem Simulation Symposium, May 1957, page 83-87.

86. Shubik, M., and Thompson, G.L., "Games of Economic Survival", NLRQ, Vol. 6, No.2, June 1959, page 111.

87. Singer, K., "Robot Economics", Economic Record, Vol.25, 1949, page 48-73.

88. Smith, N.M., "A Rationale for Operational Gaming", ORSA Meeting, Jan. 1956.

89. Speht, R.D., "Gaming as a Technique of Analysis", ORSA Meeting, Nov. 1954.

90. Stein, P.R. and Ulam, S., "A Study of Certain Combinatorial Problems Through Experiments on Computing Machines", High Speed Computer Conference, 1955.

91. Stein, P.R. and Ulam, S., "Experiments in Chess on Electronic Computing Machines", Chess Review, 1957, Page 13-17.

92. Stevenson, J.D.,MajGen., "Exercise Sagebrush", Air University Quarterly Review, Fall 1956. Page 15-38.

93. Stolzendack, G.D. and Rosenboom, J.H., "Mobilization Model for Stratigic Gaming - A Computer Application", ORSA Meeting, Nov 1956.

94. Strumpell, H.M., Maj., "Operational Simulation for Training Managers in Decision Making", U. S. A. F. Institute of Technology.

95. Taylor, J.L., "Development and Applications of a Terminal Air Battle Model", JORSA, Vol.7, No.6, Nov-Dec 1959, page 783.

96. Thomas, C.J. and Deemer, W.L., "The Role of Operational Gaming in Operations Research", JORSA, Vol.5, No.1, page 1-27.

97. Thomason, J.F., Col., "Testing High Level Plans with War Games", Army War College, March 1955.

98. Thompson, S.P. and Ziffer, A.J., "The Watchdog and the Burglar", NLRQ, Vol.6, No.2, June 1959, page 165.

99. Thompson, F.B., "War Gaming", ORSA Meeting, Nov. 1956.

100. Thrall, R.M., "Some Simple War Games", Office of Ordnance Research, Report No. 55-1, Jan 1955, page 35-54, 88-93.

101. Totten, C.A.L., "Strategos: American Games of War", D. Appleton and Co. 1880.

102. von Neumann, J., and Morgenstern, O., "Theory of Games and Economic Behaviour", Princeton University Press, 1947.

103. Walden, W., "A Study of Simple Games Through Experiments on Computing Machines", Annals of Mathematics Studies, Vol.3, No.39, 1957.

104. Walker, S.H., "Application of the Monte Carlo Method in Systems Analysis", AIIE Meeting, May 1956.

105. Weiss, H.K., "Some Differential Games of Tactical Interest and the Value of a Supporting Weapon System", JORSA, Vol.7, No.2, Mar-Apr 1959, page 180.

106. Whitson, W.L., "War Gaming at CORG", ORSA Meeting.

107. Williams, J.D., "The Compleat Strategist", McGraw-Hill Book Co., Inc., 1954.

108. Young, J.P., "A Brief History of War Gaming", CONARC, CORG-41, Oct 1955.

109. Young, J.P., "A Survey of Historical Developments in War Games", ORO/JHU, Oro-SP-98, Mar 1959.

110. Yntema, T., "Transferable Skills and Abilities", NewsLetter, Univ. Chicago, School of Business, Autumn 1957.

111. Zeuthen, F., "Problems of Monopoly and Economic Warfare", G. Routledge and Sons, London, 1930.

112. Zimmerman, R.E. and Kraft, J.F., "Carmonette: A Concept of Tactical War Games", ORO/JHU, Oro-SP-33, Nov 1957.

113. Zimmerman, R.E., "Monte Carlo Computer War Gaming", ORO/JHU, ORSA Meeting.

114. Zimmerman, R.E., "Monte Carlo Computer War Gaming, A Feasibility Study", ORO/JHU, ORO-T-325, Mar 1956.

115. Zimmerman, R.E., "A Monte Carlo Model for Military Analysis", Operations Research for Management, Vol.2, Edited by McCloskey and Coppinger, The Johns Hopkins Press, 1956.

116. Zimmerman, R.E., "Simulation of Tactical War Games", Operations Research and Systems Engineering, edited by C.D. Flagle and W.H. Huggins, and R.H. Roy, the Johns Hopkins Press, 1960.

Rand Corporation Reports, Authors unknown.

117. "An Iteration Method of Solving a Game", P-154.

118. "Simulation in Rand's System Research Laboratory", P-1074.

119. "Simulation in Rand's Logistic Systems Laboratory", P-1075.

120. "The Use of Gaming and Simulation Devices in Businees", P-1219.

121. "A Game Theory Analysis of Tactical Air War", P-1592.

122. "The Simulation of a Large Scale Military Activity", P-1555.

123. "Laboratory Evaluationof Supply and Procurement Policies", P-323.

124. "A Tank Duel with Game Theoretic Implications", T-65.

126. "A Hidden Target Model", R.M-101.

127. "Simultaneous Multiple Attacks", RM-162.

128. "Application of Theory of Games to IFF", RM-197.

129. "Total Reconnaissance with Total Countermeasures: Simplified Model", RM-202.

130. "Solutions of a Special Reconnaissance Game", RM-203.

131. "A Tactical Reconnaissance Model", RM-205.

132. "Reconnaissance in Game Theory", RM-208.

133. "Local Defense of Targets of Equal Value", RM-319, 320, 329.

134. "Local Defense versus Bombers", RM-345.

135. "n Targets of Differing Vunerability with Attacks Stronger than Defense", RM-359.

136. The Symmetrical Blotto Game", RM-424.

137. "An A - D Model", RM-463.

138. "Gaming as a Technique of Analysis", P-579.

139. "An Attack - Defense Problem for Missiles", RM-715.

140. "An Infinite - Dimensional Extension of a Symmetrical Blotto Game, RM-718.

141. "Continuous Blotto", RM-736.

142. "The Convergence of Discrete Analogues of Differential Games, Part I", RM-1480.

143. "The Simulated Aircraft and its Failure Model in LP-1", RM-2177.

144. "Cost and Performance Data from LP-1: The First Experience in Simulation by the Logistics Systems Laboratory", RM-2200.

145. "Contributions to Lanchesters Attrition Theory", RA-15078.

Authors Unknown

146. "Top Management Decision Gaming", American Management Association, May 1957.

147. "Third Symposium on Simulation and Computing Techniques Sponsored by BuAir and NADC", Oct 1953.

148. "War Gaming Handbook(U)", ATSWD-W-57-2994, Sept 1957," CONARC, SECRET.

149. "War Gaming Manual(U)", War Gaming Department, CORG, SECRET. CONARC.

150. "An Attrition Routine for the Amphibious Warfare Simulator Model", TecMemoNo K-31/59, U.S.Naval Weapons Laboratory, Dahlgren, Va.

151. "First Status Report of the Amphibious Warfare Simulation Model", TecMemoNo K-20/59, U.S.Naval Weapons Laboratory, Dahlgren, Va.

152. "Games of Strategy (Mathematics) Military Decision from the Viewpoint of Game Theory", Naval War College Review Vol.10, No.2, Oct 1957, page 27-75.

153. "Defense of the U.S. Against Attack by Aircraft and Missiles", Appendix T, "Advanced Interceptor - Missile War Game", Part II, "Machine Plays of Zigspiel(U)", Jan 1958, SECRET.

154. "Casualty Effectiveness of Toxic C.W. Munitions", "(Simplifed Procedure for Estimating Casualty Effects

HD, GB, and VX Munitions and for War Gaming)(U)", CONFIDENTIAL, June 1958, The Army Chemical Center, Maryland.

155. "War Gaming Rules", General Analysis Corp., Santa Monica, Calif., for USAEPG, Ft Huachuka, Arizona.

156. "Probability Distributions useful in War Games", Naval Research Laboratory Report 5480.

Notes.

JORSA- Journal of the Operations Research Society of America.

ORO/JHU- Operations Research Office, The Johns Hopkins University.

CORG- Combat Operations Research Group, Ft. Monroe, Va.

## APPENDIX E

## REFERENCES


Chapter I

1. Mott-Smith, Geoffrey, "Chess", Collier's Encyclopedia, 1949, Vol.5, page 92-99.

2. Weiner, M.G., "An Introduction to War Games", The Rand Corp., P-1773, Aug. 1959.

3. Young, J.P., "A Survey of Historical Developments in War Games", Operations Research Office, The Johns Hopkins University, SP-98, March 1959.

Chapter II

1. Cushen, W.E., "Generalized Battle Games on a Digital Computer", Operations Research Office, The Johns Hopkins University, T-263, Feb. 1954.

2. Meals, D. W., "Trends in Military Operations Research", Journal of the Operations Research Society of America, Vol.9, No.2, Mar-Apr. 1961.

3. Weiner, M.G., "War Gaming Methodology", The Rand Corp. RM-2413, July 1959.

4. Zimmerman, R.E., "Monte Carlo Computer War Gaming: A Feasibility Study", Operations Research Office, The Johns Hopkins University, T-325, Mar. 1956.

5. Zimmerman, R.E., "Carmonette: A Concept of Tactical War Games", Operations Research Office, The Johns Hopkins University, Jan. 1959.

Chapter III

None.

Appendix A

1. Monte Carlo Method, National Bureau of Standards Applied Mathematics Series, No. 12, U.S. Government Printing Office, Washington, D.C., 1951.

2. Symposium on Monte Carlo Methods, ed. by Herbert A. Meyer, John Wiley & Sons, New York, 1956.

3. Bradford, J.C., and Fain, W.W., Operations Research for Weapons Systems Analysis, printed at Chance Vought Aircraft, Inc., Dallas. 1959, Chapter 2.

4. Davis, P., and Rabinowitz, P., "Some Monte Carlo Experiments in Computing Multiple Integrals", <u>Mathematical Tables and Other Aids to Computation, Vol. X, No. 53, Jan. 1956.</u>

Appendix B

1. Brown, George W., "History of Rand's Random Number Digits", in <u>Monte Carlo Method</u>, National Bureau of Standards Applied Mathematics Series, No. 12, U.S. Government Printing Office, Washington, D.C., 1951, page 31-32.

2. Forsythe, G.E., "Generation and Testing of Random Digits at the National Bureau of Standards, Los Angeles", ibid., page 34-35.

3. Hammer, P.C., "The Mid-square Method of Generating Digits", ibid., page 33.

4. Johnson, D.L., "Generating and Testing Pseudo Random Numbers on the IBM 701", <u>Mathematical Tables and Other Aids to Computation</u>, Vol. X, No. 53, Jan. 1956.

5. Metropolis, N., "Phase Shifts-Middle Squares-Wave Equation", in <u>Symposium on Monte Carlo Methods</u>, ed. by Herbert A. Meyer, John Wiley & Sons, Inc., New York, 1956, page 29-36.

6. Murodch, D.C., <u>Linear Algebra for Undergraduates</u>, John Wiley & Sons, Inc., New York, 1957, page 105-108.

7. Taussky, O., and Todd, J., "Generation of Pseudo Random Numbers", <u>Symposium on Monte Carlo Methods</u>, ed. by Herbert A. Meyer, John Wiley & Sons, Inc. , New York, 1956.

8. von Neumann, J., "Various Techniques Used in Connection with Random Digits", in Monte Carlo Method, NBS AMS No. 12, U.S. GPO, Washington, D.C., 1951, pp.36-38.

9. Wall, H.S., <u>Analytic Theory of Continued Fractions</u>, D. Van Nostrand Co., Inc., New York, 1948, especially Chapter XVIII.

Appendix C

1. <u>Handbook of Automation, Computation, and Control, Vol. 2</u>, ed. by E. M. Grabbe, S. Ramo and D. E. Wooldridge, John Wiley & Sons, Inc., New York, 1959, Chapters 2, 19 and 20.

2. <u>The 1604 Computer</u>, Control Data Corporation, Minneapolis, 1959.